

Hortonworks Data Platform

Ranger Ambari Installation

(December 21, 2015)

Hortonworks Data Platform: Ranger Ambari Installation

Copyright © 2012-2015 Hortonworks, Inc. Some rights reserved.

The Hortonworks Data Platform, powered by Apache Hadoop, is a massively scalable and 100% open source platform for storing, processing and analyzing large volumes of data. It is designed to deal with data from many sources and formats in a very quick, easy and cost-effective manner. The Hortonworks Data Platform consists of the essential set of Apache Hadoop projects including MapReduce, Hadoop Distributed File System (HDFS), HCatalog, Pig, Hive, HBase, ZooKeeper and Ambari. Hortonworks is the major contributor of code and patches to many of these projects. These projects have been integrated and tested as part of the Hortonworks Data Platform release process and installation and configuration tools have also been included.

Unlike other providers of platforms built using Apache Hadoop, Hortonworks contributes 100% of our code back to the Apache Software Foundation. The Hortonworks Data Platform is Apache-licensed and completely open source. We sell only expert technical support, [training](#) and partner-enablement services. All of our technology is, and will remain, free and open source.

Please visit the [Hortonworks Data Platform](#) page for more information on Hortonworks technology. For more information on Hortonworks services, please visit either the [Support](#) or [Training](#) page. Feel free to [contact us](#) directly to discuss your specific needs.



Except where otherwise noted, this document is licensed under
Creative Commons Attribution ShareAlike 4.0 License.
<http://creativecommons.org/licenses/by-sa/4.0/legalcode>

Table of Contents

1. Overview	1
2. Installation Prerequisites	2
2.1. Configuring MySQL for Ranger	2
2.2. Configuring PostgreSQL for Ranger	3
2.3. Configuring Oracle for Ranger	4
3. Ranger Installation	6
3.1. Start the Installation	6
3.2. Customize Services	10
3.2.1. Ranger Admin Settings	11
3.2.2. Ranger Audit Settings	20
3.2.3. Configure Ranger User Sync	21
3.2.4. Configure Ranger Authentication	28
3.3. Complete the Ranger Installation	36
3.4. Configuring Ranger for LDAP SSL	37
3.4.1. Import the LDAP Cert into the Default Java TrustStore	37
3.4.2. Alternative Option	37
3.5. Setting up Database Users Without Sharing DBA Credentials	37
3.6. Updating Ranger Admin Passwords	38
4. Using Apache Solr for Ranger Audits	40
4.1. Prerequisites	40
4.2. Installing Solr	41
4.3. Configuring Solr Standalone	41
4.4. Configuring SolrCloud	42
5. Ranger Plug ins Overview	45
5.1. HDFS	45
5.2. Hive	49
5.3. HBase	53
5.4. Kafka	56
5.5. Knox	60
5.6. YARN	63
5.7. Storm	67
5.8. Manually Updating HDFS Audit Settings	71
5.9. Manually Updating Solr Audit Settings	72
6. Ranger Plugins - Kerberos Overview	74
6.1. HDFS	74
6.2. Hive	75
6.3. HBase	75
6.4. Knox	76

List of Figures

- 3.1. Installing Ranger - Main Dashboard View 6
- 3.2. Installing Ranger - Add Service 7
- 3.3. Installing Ranger - Choose Service 8
- 3.4. Installing Ranger - Ranger Requirements 9
- 3.5. Installing Ranger Assign Masters 10
- 6.1. Knox Policy Manager 77
- 6.2. Knox Repository Edit 77

List of Tables

3.1. Ranger DB Host	11
3.2. Driver Class Name	12
3.3. Ranger DB User Name Settings	12
3.4. JDBC Connect String	13
3.5. DBA Credential Settings	13
3.6. UNIX User Sync Properties	22
3.7. LDAP/AD Common Configs	24
3.8. LDAP/AD User Configs	25
3.9. LDAP/AD Group Configs	27
3.10. UNIX Authentication Settings	29
3.11. LDAP Authentication Settings	30
3.12. AD Settings	34
4.1. Solr install.properties Values	41
4.2. Solr install.properties Values	43
6.1. HDFS Plugin Properties	74
6.2. Hive Plugin Properties	75
6.3. HBase Plugin Properties	76
6.4. Knox Plugin Properties	76
6.5. Knox Configuration Properties	77

1. Overview

Apache Ranger can be installed either manually using the Hortonworks Data Platform (HDP) or the Ambari 2.1 User Interface (UI). Unlike the manual installation process, which requires you to perform a number of installation steps, installing Ranger using the Ambari UI is simpler and easier. The Ranger service option will be made available through the Add Service wizard after the HDP cluster is installed using the installation wizard.

Once Ambari has been installed and configured, you can use the Add Service wizard to install the following components:

- Ranger Admin
- Ranger UserSync
- [Ranger Key Management Service](#)

After these components are installed and started, you can enable Ranger plugins by navigating to each individual Ranger service (HDFS, HBase, Hiveserver2, Storm, Knox, YARN, and Kafka) and modifying the configuration under *advanced ranger-<service>-plugin-properties*.

Note that when you enable a Ranger plugin, you will need to restart the component.



Note

Enabling Apache Storm or Apache Kafka requires you to enable Kerberos. To enable Kerberos on your cluster, see [Enabling Kerberos Security](#) in the [Ambari Security Guide](#).

2. Installation Prerequisites

Before you install Ranger, make sure your cluster meets the following requirements:

- It is recommended that you store audits in both HDFS and Solr, so you should [install Apache Solr](#).
- To ensure that LDAP/AD group level authorization is enforced in Hadoop, you should [set up Hadoop group mapping](#) for LDAP.
- A MySQL, Oracle, PostgreSQL, MS SQL, or SQL Anywhere database instance must be running and available to be used by Ranger.

The Ranger installation will create two new users (default names: rangeradmin and rangerlogger) and two new databases (default names: ranger and ranger_audit).

- Configuration of the database instance for Ranger is described in the following sections for some of the databases supported by Ranger.
 - [Configuring MySQL for Ranger \[2\]](#)
 - [Configuring PostgreSQL for Ranger \[3\]](#)
 - [Configuring Oracle for Ranger \[4\]](#)
- If you choose not to provide system Database Administrator (DBA) account details to the Ambari Ranger installer, you can use the `dba_script.py` Python script to create Ranger DB database users without exposing DBA account information to the Ambari Ranger installer. You can then run the normal Ambari Ranger installation without specifying a DBA user name and password. For more information see [Setting up Database Users Without Sharing DBA Credentials](#).

2.1. Configuring MySQL for Ranger

1. The MySQL database administrator should be used to create the Ranger databases.

The following series of commands could be used to create the `rangerdba` user with password `rangerdba`.

- a. Log in as the root user, then use the following commands to create the `rangerdba` user and grant it adequate privileges.

```
CREATE USER 'rangerdba'@'localhost' IDENTIFIED BY 'rangerdba';
GRANT ALL PRIVILEGES ON *.* TO 'rangerdba'@'localhost';
CREATE USER 'rangerdba'@'%' IDENTIFIED BY 'rangerdba';
GRANT ALL PRIVILEGES ON *.* TO 'rangerdba'@'%';
GRANT ALL PRIVILEGES ON *.* TO 'rangerdba'@'localhost' WITH GRANT OPTION;
```

```
GRANT ALL PRIVILEGES ON *.* TO 'rangerdba'@'%' WITH GRANT OPTION;  
FLUSH PRIVILEGES;
```

- b. Use the `exit` command to exit MySQL.
- c. You should now be able to reconnect to the database as `rangerdba` using the following command:

```
mysql -u rangerdba -prangerdba
```

After testing the `rangerdba` login, use the `exit` command to exit MySQL.

2. Use the following command to confirm that the `mysql-connector-java.jar` file is in the Java share directory. This command must be run on the server where Ambari server is installed.

```
ls /usr/share/java/mysql-connector-java.jar
```

If the file is not in the Java share directory, use the following command to install the MySQL connector .jar file.

RHEL/CentOS/Oracle Linux

```
yum install mysql-connector-java*
```

SLES

```
zypper install mysql-connector-java*
```

3. Use the following command format to set the `jdbc/driver/path` based on the location of the MySQL JDBC driver .jar file. This command must be run on the server where Ambari server is installed.

```
ambari-server setup --jdbc-db={database-type} --jdbc-driver={/jdbc/driver/  
path}
```

For example:

```
ambari-server setup --jdbc-db=mysql --jdbc-driver=/usr/share/java/mysql-  
connector-java.jar
```

2.2. Configuring PostgreSQL for Ranger

1. On the PostgreSQL host, install the applicable PostgreSQL connector.

RHEL/CentOS/Oracle Linux

```
yum install postgresql-jdbc*
```

SLES

```
zypper install -y postgresql-jdbc
```

2. Confirm that the .jar file is in the Java share directory.

```
ls /usr/share/java/postgresql-jdbc.jar
```

3. Change the access mode of the .jar file to 644.

```
chmod 644 /usr/share/java/postgresql-jdbc.jar
```

4. The PostgreSQL database administrator should be used to create the Ranger databases.

The following series of commands could be used to create the rangerdba user and grant it adequate privileges.

```
echo "CREATE DATABASE $dbname;" | sudo -u $postgres psql -U postgres
echo "CREATE USER $rangerdba WITH PASSWORD '$passwd';" | sudo -u $postgres
psql -U postgres
echo "GRANT ALL PRIVILEGES ON DATABASE $dbname TO $rangerdba;" | sudo -u
postgres psql -U $postgres
```

Where:

- \$postgres is the postgres user
 - \$dbname is the name of your PostgreSQL database
5. Use the following command format to set the jdbc/driver/path based on the location of the PostgreSQL JDBC driver .jar file. This command must be run on the server where Ambari server is installed.

```
ambari-server setup --jdbc-db={database-type} --jdbc-driver={/jdbc/driver/
path}
```

For example:

```
ambari-server setup --jdbc-db=postgres --jdbc-driver=/usr/share/java/
postgresql.jar
```

6. Run the following command:

```
export HADOOP_CLASSPATH=${HADOOP_CLASSPATH}:${JAVA_JDBC_LIBS}:/connector jar
path
```

7. Add allow access details for Ranger users:

- change listen_addresses='localhost' to listen_addresses='' ('*' = any) to listen from all IPs in postgresql.conf.
- Make the following changes to the Ranger db user and Ranger audit db user in pg_hba.conf.

```
# TYPE DATABASE USER CIDR-ADDRESS METHOD
# "local" is for Unix domain socket connections only
local all postgres,rangeradmin,rangerlogger trust
# IPv4 local connections:
host all postgres,rangeradmin,rangerlogger 0.0.0.0/0 trust
# IPv6 local connections:
host all postgres,rangeradmin,rangerlogger ::/0 trust
"/var/lib/pgsql/data/pg_hba.conf" 74L, 3445C
```

2.3. Configuring Oracle for Ranger

1. On the Oracle host, install the appropriate JDBC .jar file.

- Download the Oracle JDBC (OJDBC) driver from <http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html>.
- For **Oracle Database 11g**: select Oracle Database 11g Release 2 drivers > ojdbc6.jar.
- For **Oracle Database 12c**: select Oracle Database 12c Release 1 driver > ojdbc7.jar.
- Copy the .jar file to the Java share directory. For example:

```
cp ojdbc7.jar /usr/share/java
```



Note

Make sure the .jar file has the appropriate permissions. For example:

```
chmod 644 /usr/share/java/ojdbc7.jar
```

2. The Oracle database administrator should be used to create the Ranger databases.

The following series of commands could be used to create the RANGERDBA user and grant it permissions using SQL*Plus, the Oracle database administration utility:

```
# sqlplus sys/root as sysdba
CREATE USER $RANGERDBA IDENTIFIED BY $RANGERDBAPASSWORD;
GRANT SELECT_CATALOG_ROLE TO $RANGERDBA;
GRANT CONNECT, RESOURCE TO $RANGERDBA;
QUIT;
```

3. Use the following command format to set the jdbc/driver/path based on the location of the Oracle JDBC driver .jar file. This command must be run on the server where Ambari server is installed.

```
ambari-server setup --jdbc-db={database-type} --jdbc-driver={/jdbc/driver/path}
```

For example:

```
ambari-server setup --jdbc-db=oracle --jdbc-driver=/usr/share/java/ojdbc6.jar
```

3. Ranger Installation

To install Ranger using Ambari:

1. [Start the Installation \[10 \]](#)
2. [Customize Services \[10 \]](#)
3. [Complete the Ranger Installation \[36 \]](#)

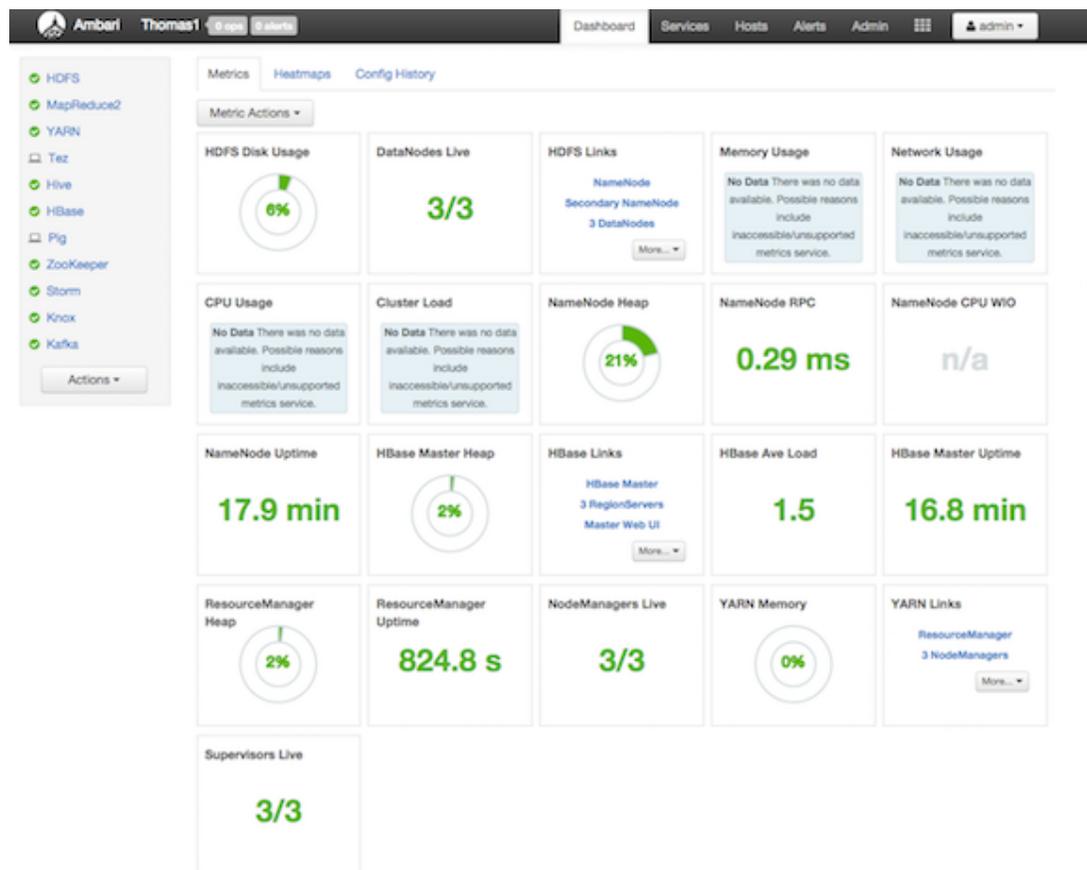
Related Topics

- [Setting up Database Users Without Sharing DBA Credentials \[37 \]](#)
- [Updating Ranger Admin Passwords \[38 \]](#)

3.1. Start the Installation

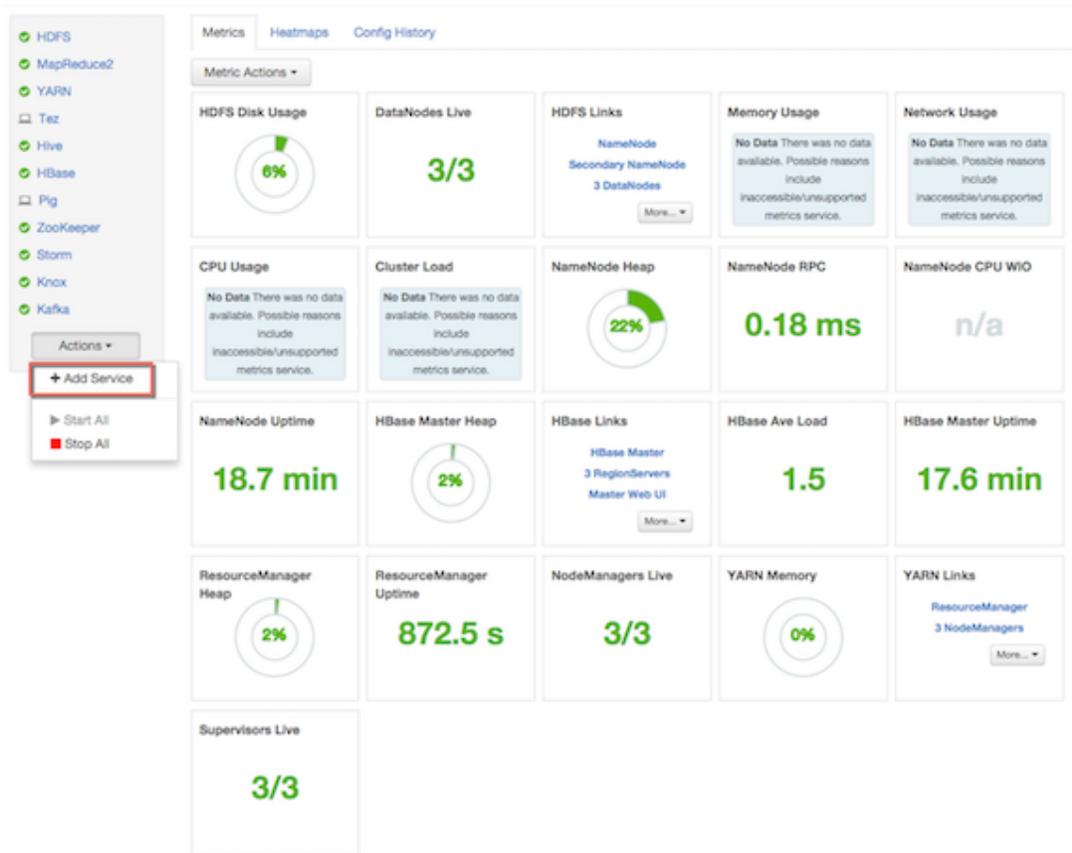
1. Log into your Ambari cluster with your designated user credentials. The main Ambari Dashboard page will be displayed.

Figure 3.1. Installing Ranger - Main Dashboard View

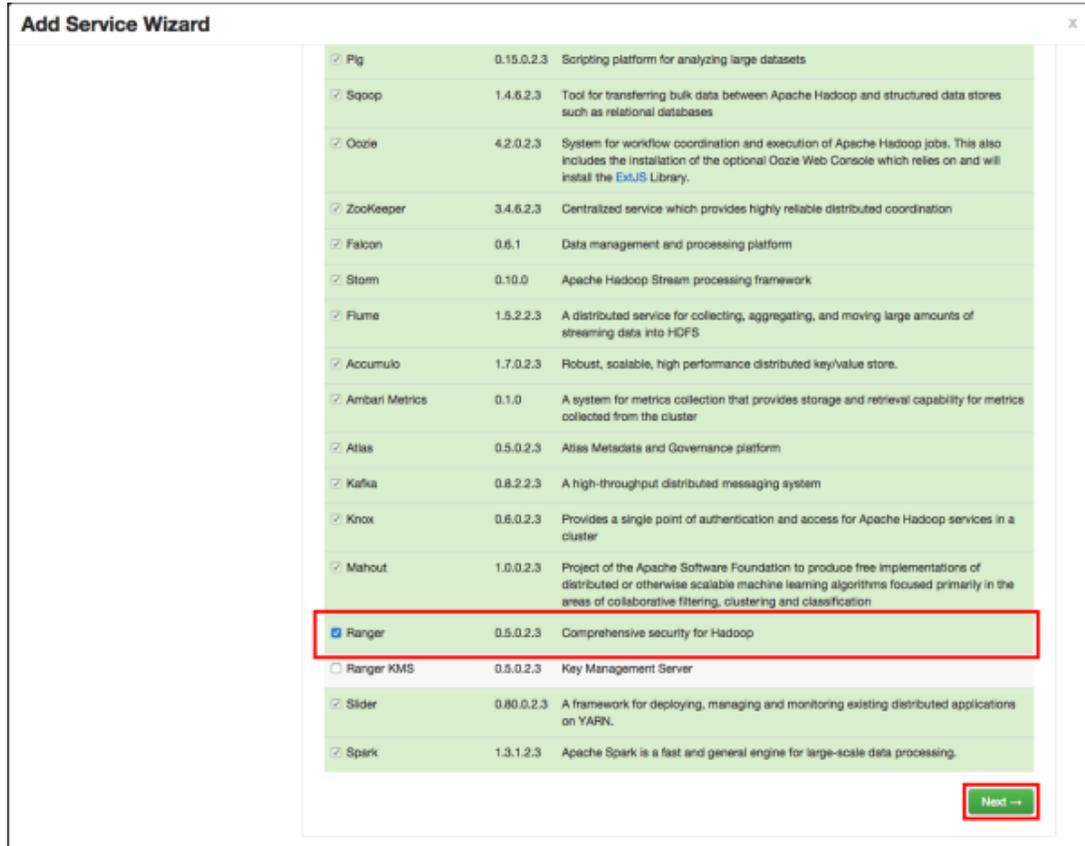


2. In the left navigation menu, click **Actions**, then select **Add Service**.

Figure 3.2. Installing Ranger - Add Service



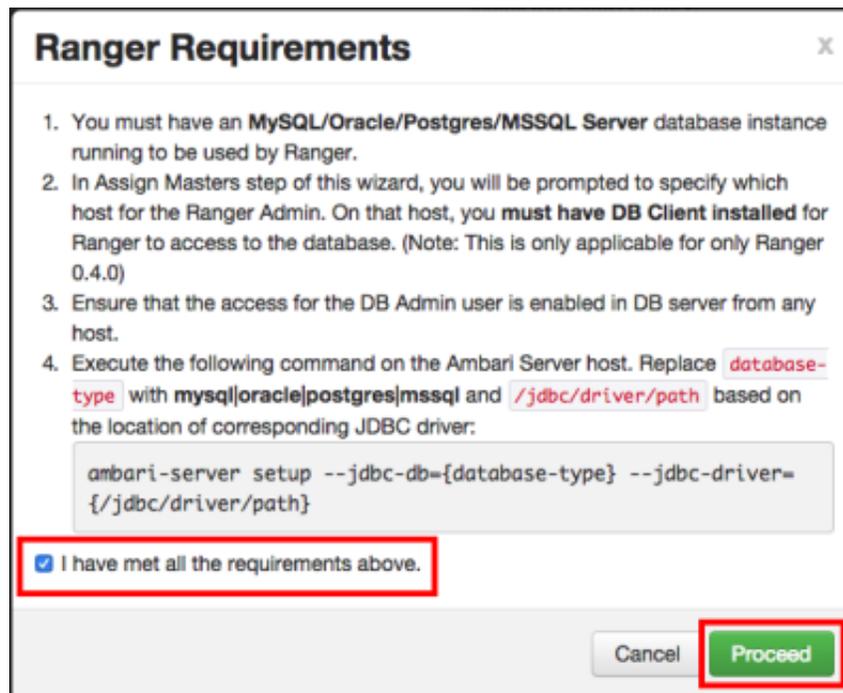
3. On the Choose Services page, select **Ranger**, then click **Next**.

Figure 3.3. Installing Ranger - Choose Service

The screenshot shows the 'Add Service Wizard' window with a list of services. The 'Ranger' service is selected and highlighted with a red box. A 'Next' button is also highlighted with a red box.

Service	Version	Description
<input checked="" type="checkbox"/> Pig	0.15.0.2.3	Scripting platform for analyzing large datasets
<input checked="" type="checkbox"/> Sqoop	1.4.6.2.3	Tool for transferring bulk data between Apache Hadoop and structured data stores such as relational databases
<input checked="" type="checkbox"/> Oozie	4.2.0.2.3	System for workflow coordination and execution of Apache Hadoop jobs. This also includes the installation of the optional Oozie Web Console which relies on and will install the ExUS Library.
<input checked="" type="checkbox"/> ZooKeeper	3.4.6.2.3	Centralized service which provides highly reliable distributed coordination
<input checked="" type="checkbox"/> Falcon	0.6.1	Data management and processing platform
<input checked="" type="checkbox"/> Storm	0.10.0	Apache Hadoop Stream processing framework
<input checked="" type="checkbox"/> Flume	1.5.2.2.3	A distributed service for collecting, aggregating, and moving large amounts of streaming data into HDFS
<input checked="" type="checkbox"/> Accumulo	1.7.0.2.3	Robust, scalable, high performance distributed key/value store.
<input checked="" type="checkbox"/> Ambari Metrics	0.1.0	A system for metrics collection that provides storage and retrieval capability for metrics collected from the cluster
<input checked="" type="checkbox"/> Atlas	0.5.0.2.3	Atlas Metadata and Governance platform
<input checked="" type="checkbox"/> Kafka	0.8.2.2.3	A high-throughput distributed messaging system
<input checked="" type="checkbox"/> Knox	0.6.0.2.3	Provides a single point of authentication and access for Apache Hadoop services in a cluster
<input checked="" type="checkbox"/> Mahout	1.0.0.2.3	Project of the Apache Software Foundation to produce free implementations of distributed or otherwise scalable machine learning algorithms focused primarily in the areas of collaborative filtering, clustering and classification
<input checked="" type="checkbox"/> Ranger	0.5.0.2.3	Comprehensive security for Hadoop
<input type="checkbox"/> Ranger KMS	0.5.0.2.3	Key Management Server
<input checked="" type="checkbox"/> Slider	0.80.0.2.3	A framework for deploying, managing and monitoring existing distributed applications on YARN.
<input checked="" type="checkbox"/> Spark	1.3.1.2.3	Apache Spark is a fast and general engine for large-scale data processing.

4. The Ranger Requirements page appears. Ensure that you have met all of the installation requirements, then select the "I have met all the requirements above" check box and click **Proceed**.

Figure 3.4. Installing Ranger - Ranger Requirements

5. You are then prompted to select the host where Ranger Admin will be installed. This host should have DB admin access to the Ranger DB host and User Sync. Notice in the figure below that both the Ranger Admin and Ranger User Sync services will be installed on the primary node in the cluster (c6401.ambari.apache.org in the example shown below).

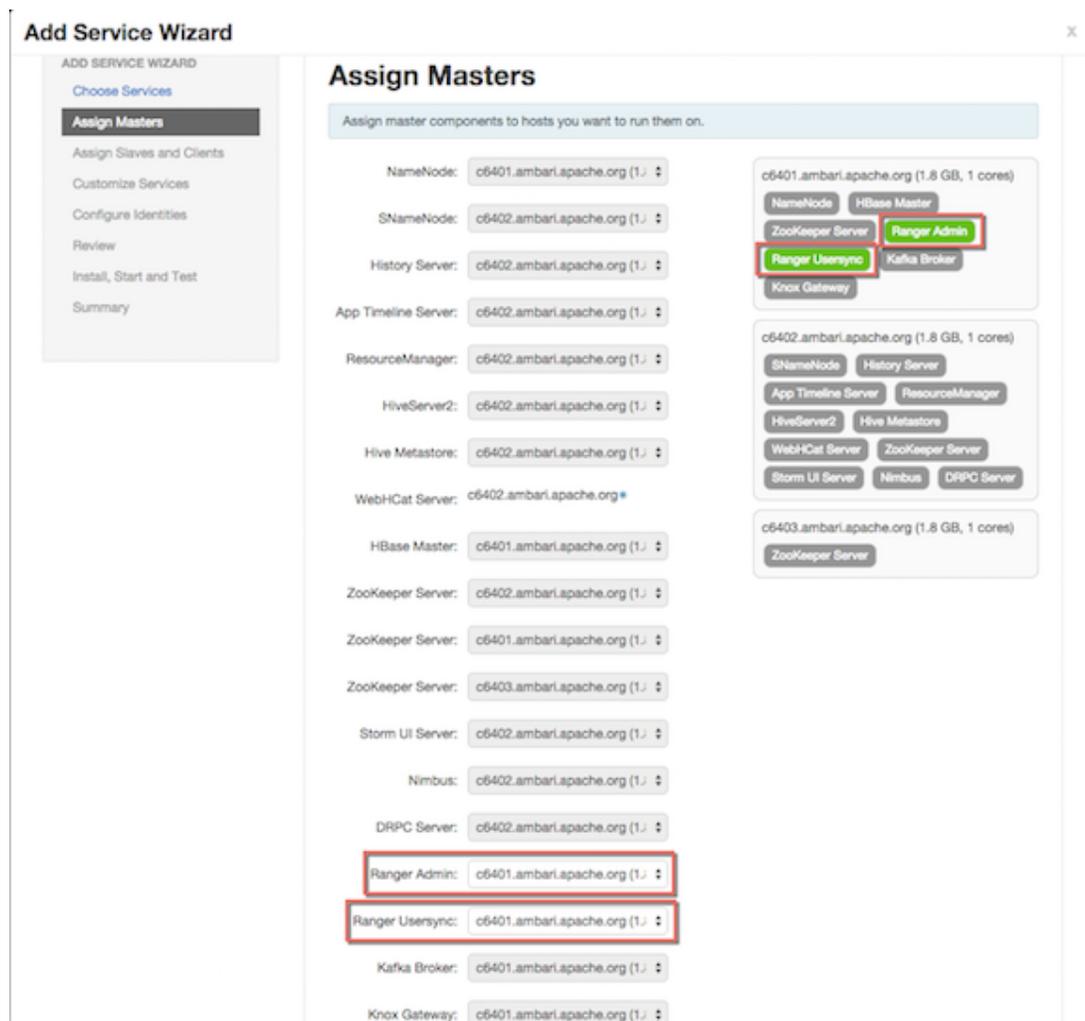
Make a note of the Ranger Admin host for use in subsequent installation steps. Click **Next** when finished to continue with the installation.



Note

The Ranger Admin and Ranger User Sync services must be installed on the same cluster node.

Figure 3.5. Installing Ranger Assign Masters



6. The Customize Services page appears. These settings are described in the next section.

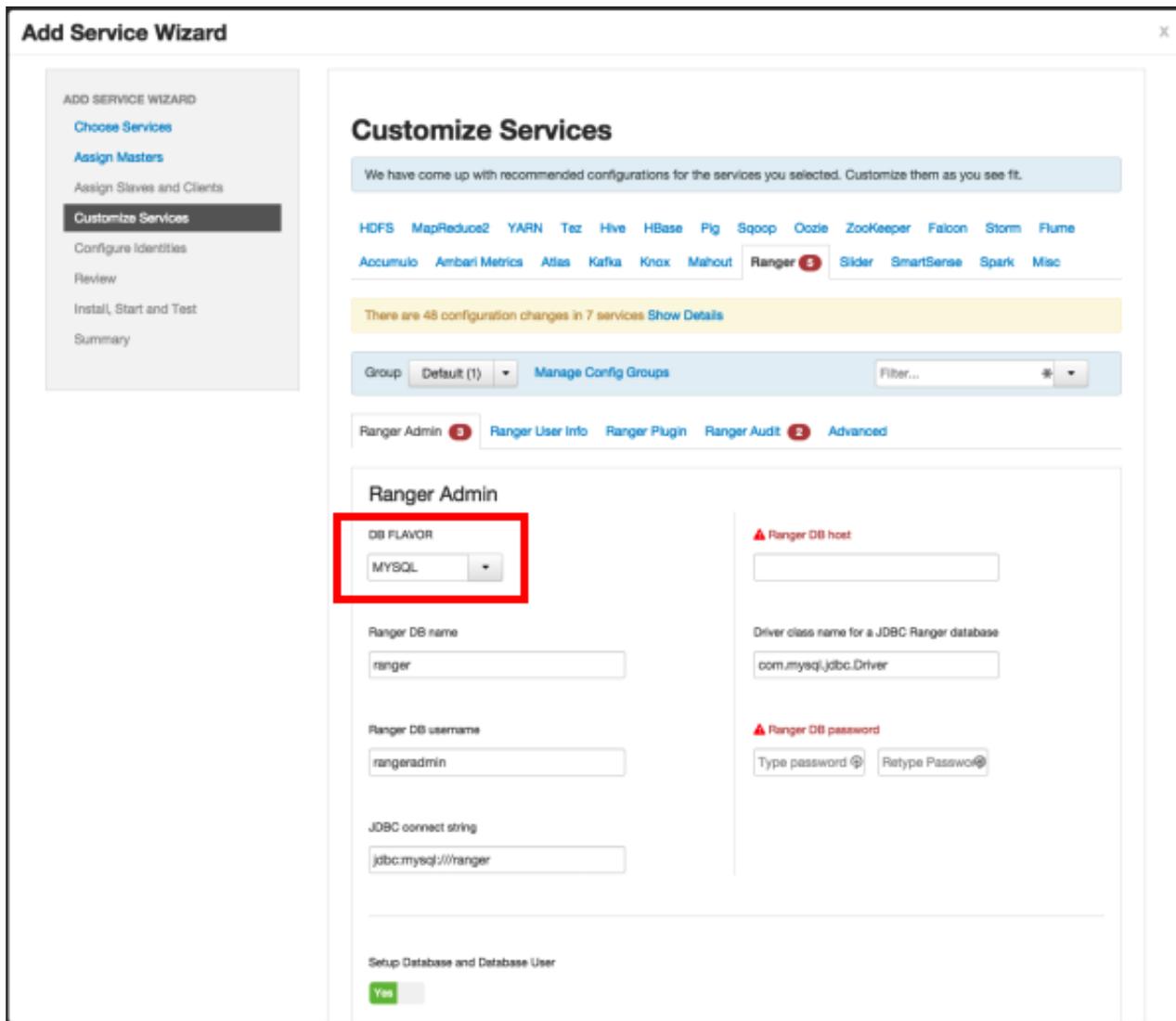
3.2. Customize Services

The next step in the installation process is to specify Ranger settings on the Customize Services page.

- [Ranger Admin Settings \[11\]](#)
- [Ranger Audit Settings \[20\]](#)
- [Configure Ranger User Sync \[21\]](#)
- [Configure Ranger Authentication \[28\]](#)

3.2.1. Ranger Admin Settings

1. On the Customize Services page, select the Ranger Admin tab, then use the **DB Flavor** drop-down to select the database type that you are using with Ranger.



2. Enter the database server address in the **Ranger DB Host** box.

Table 3.1. Ranger DB Host

DB Flavor	Host	Example
MySQL	<HOST[:PORT]>	c6401.ambari.apache.org or c6401.ambari.apache.org:3306
	<HOST:PORT:SID>	c6401.ambari.apache.org:1521:ORCL
Oracle	<HOST:PORT/Service>	c6401.ambari.apache.org:1521/XE
PostgreSQL	<HOST[:PORT]>	c6401.ambari.apache.org

DB Flavor	Host	Example
		or c6401.ambari.apache.org:5432
MS SQL	<HOST[:PORT]>	c6401.ambari.apache.org or c6401.ambari.apache.org:1433
SQLA	<HOST[:PORT]>	c6401.ambari.apache.org or c6401.ambari.apache.org:2638

- Ranger DB name** – The name of the Ranger Policy database, i.e. ranger_db. Please note that if you are using Oracle, you must specify the Oracle tablespace name here.
- Driver class name for a JDBC Ranger database – the driver class name is automatically generated based on the selected DB Flavor. The table below lists the default driver class settings. Currently Ranger does not support any third party JDBC driver.

Table 3.2. Driver Class Name

DB Flavor	Driver class name for a JDBC Ranger database
MySQL	com.mysql.jdbc.Driver
Oracle	oracle.jdbc.driver.OracleDriver
PostgreSQL	org.postgresql.Driver
MS SQL	com.microsoft.sqlserver.jdbc.SQLServerDriver
SQLA	sap.jdbc4.sqlanywhere.IDriver

- Ranger DB username and Ranger DB Password** – Enter the user name and passwords for your Ranger database server. The following table describes these settings in more detail. You can use the MySQL database that was installed with Ambari, or an external MySQL, Oracle, PostgreSQL, MS SQL or SQL Anywhere database.

Table 3.3. Ranger DB User Name Settings

Property	Description	Default Value	Example Value	Required?
Ranger DB username	The username for the Policy database.	rangeradmin	rangeradmin	Yes
Ranger DB password	The password for the Ranger Policy database user.		PassWORD	Yes

6. JDBC connect string



Important

Currently the Ambari installer generates the JDBC connect string using the `jdbc:oracle:thin:@//host:port/db_name` format. You must replace the connection string as described in the following table:

Table 3.4. JDBC Connect String

DB Flavor	Syntax	Example Value
MySQL	jdbc:mysql://DB_HOST:PORT/ db_name	jdbc:mysql:// c6401.ambari.apache.org:3306/ ranger_db
Oracle	For Oracle SID: jdbc:oracle:thin:@DB_HOST:PORT:SID	jdbc:oracle:thin:@c6401.ambari.apache.org:1521:ORCL
	For Oracle Service Name: jdbc:oracle:thin:@//DB_HOST[:PORT] [/ServiceName]	jdbc:oracle:thin:@// c6401.ambari.apache.org:1521/XE
PostgreSQL	jdbc:postgresql://DB_HOST/ db_name	jdbc:postgresql:// c6401.ambari.apache.org:5432/ ranger_db
MS SQL	jdbc:sqlserver:// DB_HOST;databaseName=db_name	jdbc:sqlserver:// c6401.ambari.apache.org:1433;databaseName=ranger_db
SQLA	jdbc:sqlanywhere:host=DB_HOST;data	jdbc:sqlanywhere:host=c6401.ambari.apache.org:2638;data

7. Setup Database and Database User

- If set to **Yes** – The Database Administrator (DBA) user name and password will need to be provided as described in the next step.



Note

Ranger does not store the DBA user name and password after setup. Therefore, you can clear these values in the Ambari UI after the Ranger setup is complete.

- If set to **No** – A **No** indicates that you do not wish to provide Database Administrator (DBA) account details to the Ambari Ranger installer. Setting this to No continues the Ranger installation process without providing DBA account details. In this case, you must perform the system database user setup as described in [Setting up Database Users Without Sharing DBA Credentials](#), and then proceed with the installation.



Note

If **No** is selected and the UI still requires you to enter a user name and password in order to proceed, you can enter any value – the values do not need to be the actual DBA user name and password.

- 8. Database Administrator (DBA) username and Database Administrator (DBA) password** – The DBA username and password are set when the database server is installed. If you do not have this information, contact the database administrator who installed the database server.

Table 3.5. DBA Credential Settings

Property	Description	Default Value	Example Value	Required?
Database Administrator (DBA) username	The Ranger database user that has administrative	root	root	Yes

Property	Description	Default Value	Example Value	Required?
	privileges to create database schemas and users.			
Database Administrator (DBA) password	The root password for the Ranger database user.		root	Yes

If the Oracle DB root user Role is SYSDBA, you must also specify that in the **Database Administrator (DBA) username** parameter. For example, if the DBA user name is `orcl_root` you must specify `orcl_root AS SYSDBA`.



Note

As mentioned in the note in the previous step, if **Setup Database and Database User** is set to **No**, a placeholder DBA username and password may still be required in order to continue with the Ranger installation.

The following images show examples of the DB settings for each Ranger database type.



Note

To test the DB settings, click **Test Connection**. If a Ranger database has not been pre-installed, **Test Connection** will fail even for a valid configuration.

MySQL

Ranger Admin [Ranger User Info](#) [Ranger Plugin](#) [Ranger Audit](#) [Advanced](#)

Ranger Admin

<p>DB FLAVOR</p> <input type="text" value="MYSQL"/>	<p>Ranger DB host</p> <input type="text" value="c6401.ambari.apache.org"/>
<p>Ranger DB name</p> <input type="text" value="ranger"/>	<p>Driver class name for a JDBC Ranger database</p> <input type="text" value="com.mysql.jdbc.Driver"/>
<p>Ranger DB username</p> <input type="text" value="rangeradmin"/>	<p>Ranger DB password</p> <input type="password" value="....."/> <input type="password" value="....."/>
<p>JDBC connect string</p> <input type="text" value="jdbc:mysql://c6401.ambari.apache.org/ranger"/>	

Setup Database and Database User

Yes

<p>Database Administrator (DBA) username</p> <input type="text" value="root"/>	<p>Database Administrator (DBA) password</p> <input type="password" value="....."/> <input type="password" value="....."/>
<p>JDBC connect string for root user</p> <input type="text" value="jdbc:mysql://c6401.ambari.apache.org"/>	

Oracle – if the Oracle instance is running with a Service name.

Ranger Admin | Ranger User Info | Ranger Plugin | Ranger Audit | Advanced

Ranger Admin

DB FLAVOR

Ranger DB name

Ranger DB username

JDBC connect string

Ranger DB host

Driver class name for a JDBC Ranger database

Ranger DB password

Setup Database and Database User
 Yes

Database Administrator (DBA) username

Database Administrator (DBA) password

JDBC connect string for root user
 

Oracle – if the Oracle instance is running with a SID.

Ranger Admin [Ranger User Info](#) [Ranger Plugin](#) [Ranger Audit](#) [Advanced](#)

Ranger Admin

DB FLAVOR

Ranger DB name

Ranger DB username

JDBC connect string

Ranger DB host

Driver class name for a JDBC Ranger database

Ranger DB password

Setup Database and Database User
 Yes

Database Administrator (DBA) username

JDBC connect string for root user

Database Administrator (DBA) password

PostgreSQL

Ranger Admin [Ranger User Info](#) [Ranger Plugin](#) [Ranger Audit](#) [Advanced](#)

Ranger Admin

DB FLAVOR

Ranger DB name

Ranger DB username

JDBC connect string

Ranger DB host

Driver class name for a JDBC Ranger database

Ranger DB password

Setup Database and Database User
 Yes

Database Administrator (DBA) username

JDBC connect string for root user

Database Administrator (DBA) password

MS SQL

Ranger Admin [Ranger User Info](#) [Ranger Plugin](#) [Ranger Audit](#) [Advanced](#)

Ranger Admin

DB FLAVOR
MSSQL

Ranger DB name
ranger

Ranger DB username
rangeradmin

JDBC connect string
r1.ambari.apache.org:1433;databaseName=ranger

Ranger DB host
c6401.ambari.apache.org:1433

Driver class name for a JDBC Ranger database
com.microsoft.sqlserver.jdbc.SQLServerDriver

Ranger DB password
.....

Setup Database and Database User
 Yes

Database Administrator (DBA) username
administrator

JDBC connect string for root user
jdbc:sqlserver://c6401.ambari.apache.org:1433;

Database Administrator (DBA) password
.....

SQL Anywhere

Ranger Admin **Ranger User Info** Ranger Plugin Ranger Audit Advanced

Ranger Admin

DB FLAVOR
SQL Anywhere ▼

Ranger DB name
ranger

Ranger DB username
rangeradmin

JDBC connect string
=c6401.ambari.apache.org:2638;database=ranger

Ranger DB host
c6401.ambari.apache.org:2638

Driver class name for a JDBC Ranger database
sap.jdbc4.sqlanywhere.IDriver

Ranger DB password
.....

Setup Database and Database User
 Yes

Database Administrator (DBA) username
dba

JDBC connect string for root user
jdbc:sqlanywhere:host=c6401.ambari.apache.org::

Database Administrator (DBA) password
.....

3.2.2. Ranger Audit Settings

1. On the Customize Services page, select the Ranger Audit tab.

It is recommended that you store audits in Solr and HDFS, and disable Audit to DB.

2. Under Audit to Solr, enter the Solr audit URL in the **ranger.audit.solr.urls** box using the following format:

```
http://<solr_host>:6083/solr/ranger_audits
```



Note

Audits to Solr requires that you have already [installed Solr](#).

3. Under Audit to DB, enter a password in the **Ranger Audit DB password** boxes. Audit to DB is set to **Off** by default. (The password must be entered to preserve backward compatibility)

3.2.3. Configure Ranger User Sync

This section describes how to configure Ranger User Sync for either UNIX or LDAP/AD.

- [Configuring Ranger User Sync for UNIX \[22\]](#)
- [Configuring Ranger User Sync for LDAP/AD \[23\]](#)

3.2.3.1. Configuring Ranger User Sync for UNIX

Use the following steps to configure Ranger User Sync for UNIX.

1. On the Customize Services page, select the Ranger User Info tab.
2. Click **Yes** under Enable User Sync.
3. Use the Sync Source drop-down to select UNIX, then set the following properties.

Table 3.6. UNIX User Sync Properties

Property	Description	Default Value
Sync Source	Only sync users above this user ID.	500
Password File	The location of the password file on the Linux server.	/etc/passwd
Group File	The location of the groups file on the Linux server.	/etc/group

Add Service Wizard

We have come up with recommended configurations for the services you selected. Customize them as you see fit.

HDFS MapReduce2 YARN Tez Hive HBase Pig Sqoop Oozie ZooKeeper Falcon Storm Flume
Accumulo Ambari Metrics Atlas Kafka Knox Mahout **Ranger** Slider SmartSense Spark Misc

There are 50 configuration changes in 8 services. [Show Details](#)

Group: Default (1) Manage Config Groups Filter...

Ranger Admin Ranger User Info Ranger Plugin Ranger Audit Advanced

Ranger User Info

Enable User Sync
 Yes

Sync Source
LDAP

Minimum User ID
500

Password File
/etc/passwd

Group File
/etc/group

All configurations have been addressed.

3.2.3.2. Configuring Ranger User Sync for LDAP/AD



Important

To ensure that LDAP/AD group level authorization is enforced in Hadoop, you should [set up Hadoop group mapping for LDAP/AD](#).



Note

You can use the [LDAP Connection Check tool](#) to determine User Sync settings for LDAP/AD.

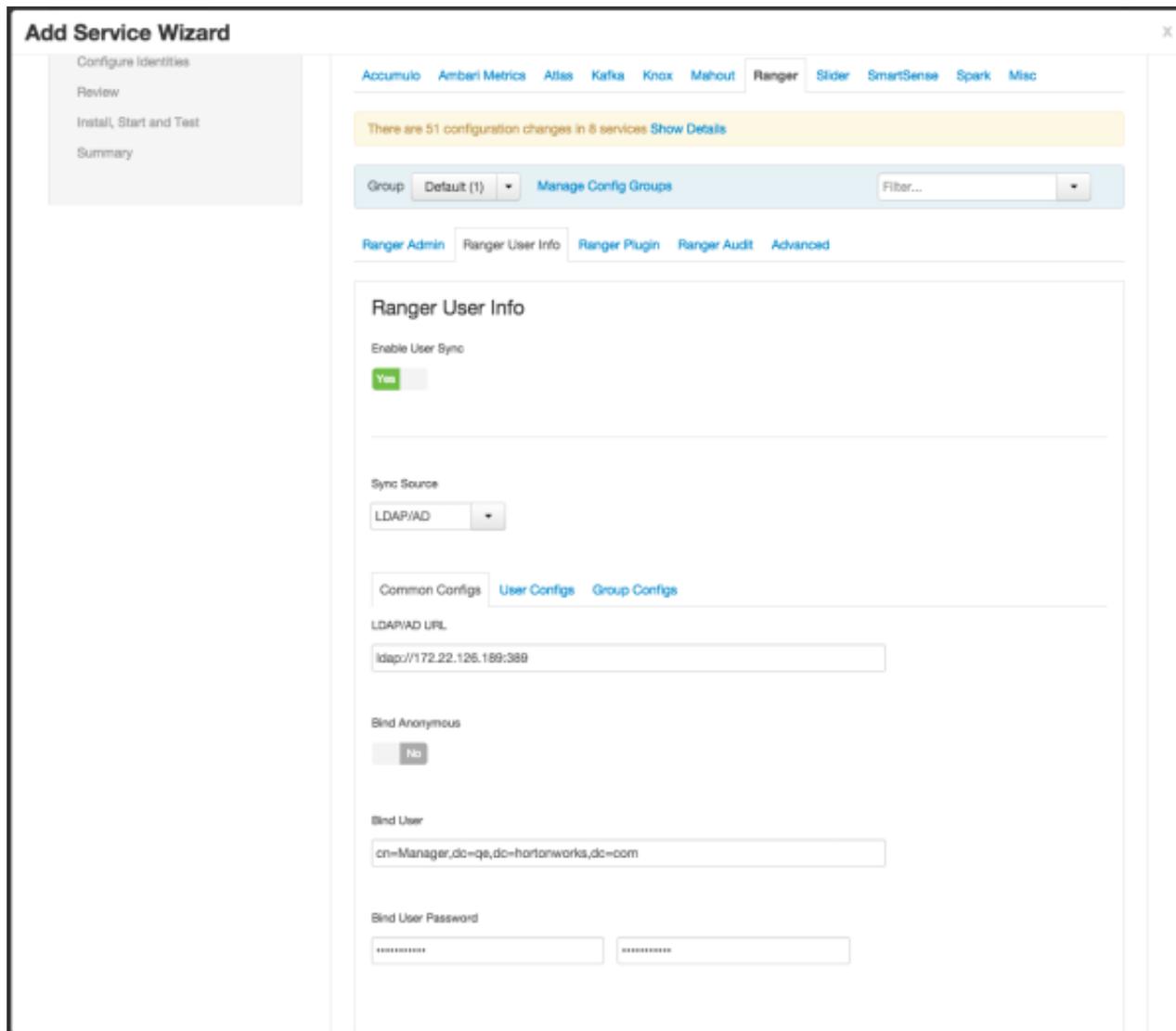
Use the following steps to configure Ranger User Sync for LDAP/AD.

1. On the Customize Services page, select the Ranger User Info tab.

2. Click **Yes** under Enable User Sync.
3. Use the Sync Source drop-down to select LDAP/AD.
4. Set the following properties on the Common Configs tab.

Table 3.7. LDAP/AD Common Configs

Property	Description	Default Value	Sample Values
LDAP/AD URL	Add URL depending upon LDAP/AD sync source	ldap://{host}:{port}	ldap:// ldap.example.com:389 or ldaps:// ldap.example.com:636
Bind Anonymous	If Yes is selected, the Bind User and Bind User Password are not required.	NO	
Bind User	The location of the groups file on the Linux server.	The full distinguished name (DN), including common name (CN), of an LDAP/AD user account that has privileges to search for users. The LDAP bind DN is used to connect to LDAP and query for users and groups.	cn=admin,dc=example,dc=com or admin@example.com
Bind User Password	The password of the Bind User.		



5. Set the following properties on the User Configs tab.

Table 3.8. LDAP/AD User Configs

Property	Description	Default Value	Sample Values
Group User Map Sync	Sync specific groups for users.	No	Yes
Username Attribute	The LDAP user name attribute.		sAMAccountName for AD, uid or cn for OpenLDAP
User Object Class	Object class to identify user entries.	person	top, person, organizationalPerson, user, or posixAccount
User Search Base	Search base for users.		cn=users,dc=example,dc=com
User Search Filter	Optional additional filter constraining the users selected for syncing.		Sample filter to retrieve all the users: cn= Sample filter to retrieve all the users who are members

Property	Description	Default Value	Sample Values
			of groupA or groupB: ((memberof=CN=GroupA,OU=groups,DC=example.com) (memberof=CN=GroupB,OU=groups,DC=example.com)
User Search Scope	This value is used to limit user search to the depth from search base.	sub	base, one, or sub
User Group Name Attribute	Attribute from user entry whose values would be treated as group values to be pushed into the Policy Manager database. You can provide multiple attribute names separated by commas.	memberof,ismemberof	memberof, ismemberof, or gidNumber

Add Service Wizard

Ranger User Info

Enable User Sync
 Yes

Sync Source
 LDAP/AD

Common Configs | **User Configs** | Group Configs

Group User Map Sync
 Yes

Username Attribute

User Object Class

User Search Base

User Search Filter

User Search Scope

User Group Name Attribute

6. Set the following properties on the Group Configs tab.

Table 3.9. LDAP/AD Group Configs

Property	Description	Default Value	Sample Values
Enable Group Sync	<p>If Enable Group Sync is set to No, the group names the users belong to are derived from "User Group Name Attribute". In this case no additional group filters are applied.</p> <p>If Enable Group Sync is set to Yes, the groups the users belong to are retrieved from LDAP/AD using the following group-related attributes.</p>	No	Yes
Group Member Attribute	The LDAP group member attribute name.		member
Group Name Attribute	The LDAP group name attribute.		distinguishedName for AD, cn for OpenLdap
Group Object Class	LDAP Group object class.		group, groupofnames, or posixGroup
Group Search Base	Search base for groups.		ou=groups,DC=example,DC=com
Group Search Filter	Optional additional filter constraining the groups selected for syncing.		<p>Sample filter to retrieve all groups: cn=*</p> <p>Sample filter to retrieve only the groups whose cn is Engineering or Sales: (& (cn=Engineering)(cn=Sales))</p>

The screenshot shows the 'Add Service Wizard' interface with the 'Ranger User Info' tab selected. The 'Enable User Sync' toggle is set to 'Yes'. The 'Sync Source' dropdown is set to 'LDAP/AD'. Below this, there are three sub-tabs: 'Common Configs', 'User Configs', and 'Group Configs', with 'Group Configs' being the active one. Under 'Group Configs', the following fields are visible: 'Enable Group Sync' (set to 'Yes'), 'Group Member Attribute' (value: 'member'), 'Group Name Attribute' (value: 'cn'), 'Group Object Class' (value: 'groupOfNames'), 'Group Search Base' (value: 'dc=qa,dc=hortonworks,dc=com'), and 'Group Search Filter' (value: 'gp=*').

3.2.4. Configure Ranger Authentication

This section describes how to configure Ranger authentication for UNIX, LDAP, and AD.

- [Configuring Ranger UNIX Authentication \[28\]](#)
- [Configuring Ranger LDAP Authentication \[30\]](#)
- [Configuring Ranger Active Directory Authentication \[33\]](#)

3.2.4.1. Configuring Ranger UNIX Authentication

Use the following steps to configure Ranger authentication for UNIX.

1. Select the Advanced tab on the Customize Services page.

2. Under Ranger Settings, specify the Ranger Policy Manager host address in the **External URL** box in the format `http://<your_ranger_host>:6080`.
3. Under Ranger Settings, select **UNIX**.
 HTTP is enabled by default – if you disable HTTP, only HTTPS is allowed.
4. Under UNIX Authentication Settings, set the following properties.

Table 3.10. UNIX Authentication Settings

Property	Description	Default Value	Example Value
Allow remote Login	Flag to enable/disable remote login. Only applies to UNIX authentication.	true	true
ranger.unixauth.service.hostname	The address of the host where the UNIX authentication service is running.	{{ugsync_host}}	{{ugsync_host}}
ranger.unixauth.service.port	The port number on which the UNIX authentication service is running.	5151	5151



Note

Properties with value `{{xyz}}` are macro variables that are derived from other specified values in order to streamline the configuration process. Macro variables can be edited if required – if you need to restore the original value, click the Set Recommended symbol at the right of the property box.

Add Service Wizard

Ranger Settings

External URL:

Authentication method:
 LDAP
 ACTIVE_DIRECTORY
 UNIX
 NONE

HTTP enabled:

Unix Authentication Settings

Allow remote Login:

ranger.unixauth.service.hostname:

ranger.unixauth.service.port:

Knox SSO Settings

Advanced ranger-admin-site

3.2.4.2. Configuring Ranger LDAP Authentication



Note

You can use the [LDAP Connection Check tool](#) to determine authentication settings for LDAP.

Use the following steps to configure Ranger authentication for LDAP.

1. Select the Advanced tab on the Customize Services page.
2. Under Ranger Settings, specify the Ranger Policy Manager host address in the **External URL** box in the format `http://<your_ranger_host>:6080`.
3. Under Ranger Settings, select **LDAP**.
4. Under LDAP Settings, set the following properties.

Table 3.11. LDAP Authentication Settings

Property	Description	Default Value	Example Value
ranger.ldap.base.dn	The Distinguished Name (DN) of the starting point for directory server searches.	dc=example,dc=com	dc=example,dc=com
Bind User	The full Distinguished Name (DN), including Common Name (CN) of an LDAP user account that has privileges to search for users. This is a macro variable value that is derived from the Bind User value from Ranger User Info > Common Configs .	{{ranger_ug_ldap_bind_dn}}ranger_ug_ldap_bind_dn}}	
Bind User Password	Password for the Bind User. This is a macro variable value that is derived from the Bind User Password value from Ranger User Info		

Property	Description	Default Value	Example Value
	> Common Configs.		
ranger.ldap.group.roleattribute	The LDAP group role attribute.	cn	cn
ranger.ldap.referral	See description below.	ignore	follow ignore throw
LDAP URL	The LDAP server URL. This is a macro variable value that is derived from the LDAP/AD URL value from Ranger User Info > Common Configs.	{{ranger_ug_ldap_url}}	{{ranger_ug_ldap_url}}
ranger.ldap.user.dnpattern	The user DN pattern is expanded when a user is being logged in. For example, if the user "ldadmin" attempted to log in, the LDAP Server would attempt to bind against the DN "uid=ldadmin,ou=users,dc=example,dc=com" using the password the user provided>	uid={0},ou=users,dc=xasecure,dc=net	cn=ldadmin,ou=Users,dc=example,dc=com
User Search Filter	The search filter used for Bind Authentication. This is a macro variable value that is derived from the User Search Filter value from Ranger User Info > User Configs.	{{ranger_ug_ldap_user_searchfilter}}	{{ranger_ug_ldap_user_searchfilter}}



Note

Properties with value `{{xyz}}` are macro variables that are derived from other specified values in order to streamline the configuration process. Macro variables can be edited if required – if you need to restore the original value, click the Set Recommended symbol at the right of the property box.

There are three possible values for `ranger.ldap.referral`: `follow`, `throw`, and `ignore`. The recommended setting is `follow`.

When searching a directory, the server might return several search results, along with a few continuation references that show where to obtain further results. These results and references might be interleaved at the protocol level.

- When this property is set to `follow`, the LDAP service provider processes all of the normal entries first, and then follows the continuation references.
- When this property is set to `throw`, all of the normal entries are returned in the enumeration first, before the `ReferralException` is thrown. By contrast, a "referral" error response is processed immediately when this property is set to `follow` or `throw`.
- When this property is set to `ignore`, it indicates that the server should return referral entries as ordinary entries (or plain text). This might return partial results for the search.

3.2.4.3. Configuring Ranger Active Directory Authentication



Note

You can use the [LDAP Connection Check tool](#) to determine authentication settings for Active Directory.

Use the following steps to configure Ranger authentication for Active Directory.

1. Select the Advanced tab on the Customize Services page.
2. Under Ranger Settings, specify the Ranger Policy Manager host address in the **External URL** box in the format `http://<your_ranger_host>:6080`.
3. Under Ranger Settings, select **ACTIVE_DIRECTORY**.
4. Under AD Settings, set the following properties.

Table 3.12. AD Settings

Property	Description	Default Value	Example Value
ranger.ldap.ad.base.dn	The Distinguished Name (DN) of the starting point for directory server searches.	dc=example,dc=com	dc=example,dc=com
ranger.ldap.ad.bind.dn	The full Distinguished Name (DN), including Common Name (CN) of an LDAP user account that has privileges to search for users. This is a macro variable value that is derived from the Bind User value from Ranger User Info > Common Configs .	{{ranger_ug_ldap_bind_dn}}	{{ranger_ug_ldap_bind_dn}}
ranger.ldap.ad.bind.password	Password for the bind.dn. This is a macro variable value that is derived from the Bind User Password value from Ranger User Info > Common Configs .		
Domain Name (Only for AD)	The domain name of the AD Authentication service.		dc=example,dc=com
ranger.ldap.ad.referral	See description below.	ignore	follow ignore throw
ranger.ldap.ad.url	The AD server URL. This is a macro variable value that is derived from the LDAP/AD URL value from Ranger User Info > Common Configs .	{{ranger_ug_ldap_url}}	{{ranger_ug_ldap_url}}
ranger.ldap.ad.user.searchfilter	The search filter used for Bind Authentication. This is a macro variable value that is derived from the User Search Filter value from Ranger User Info > User Configs .	{{ranger_ug_ldap_user_searchfilter}}	{{ranger_ug_ldap_user_searchfilter}}



Note

Properties with value `{{xyz}}` are macro variables that are derived from other specified values in order to streamline the configuration process. Macro variables can be edited if required – if you need to restore the original value, click the Set Recommended symbol at the right of the property box.

There are three possible values for `ranger.ldap.ad.referral`: `follow`, `throw`, and `ignore`. The recommended setting is `follow`.

When searching a directory, the server might return several search results, along with a few continuation references that show where to obtain further results. These results and references might be interleaved at the protocol level.

- When this property is set to `follow`, the AD service provider processes all of the normal entries first, and then follows the continuation references.

- When this property is set to `throw`, all of the normal entries are returned in the enumeration first, before the `ReferralException` is thrown. By contrast, a "referral" error response is processed immediately when this property is set to `follow` or `throw`.
- When this property is set to `ignore`, it indicates that the server should return referral entries as ordinary entries (or plain text). This might return partial results for the search. In the case of AD, a `PartialResultException` is returned when referrals are encountered while search results are processed.

The screenshot shows the 'Add Service Wizard' dialog box. It has two main sections: 'Ranger Settings' and 'AD Settings'. In the 'Ranger Settings' section, 'External URL' is set to 'http://05403.ambari.apache.org:6080', 'Authentication method' is set to 'ACTIVE_DIRECTORY' (selected with a radio button), and 'HTTP enabled' is checked. In the 'AD Settings' section, 'ranger.ldap.ad.base.dn' is 'dc=example,dc=com', 'ranger.ldap.ad.bind.dn' is '{{ranger_ug_ldap_bind_dn}}', 'ranger.ldap.ad.bind.password' is masked with dots, 'Domain Name (Only for AD)' is 'dc=hwqe,dc=hortonworks,dc=com', 'ranger.ldap.ad.referral' is 'ignore', 'ranger.ldap.ad.uri' is '{{ranger_ug_ldap_uri}}', and 'ranger.ldap.ad.user.searchfilter' is '{{ranger_ug_ldap_user_searchfilter}}'. There are also sections for 'Knox SSO Settings' and 'Advanced ranger-admin-site' which are currently collapsed.

When you have finished configuring all of the Customize Services Settings, click **Next** at the bottom of the page to continue with the installation.

5. When you save the authentication method as Active Directory, a Dependent Configurations pop-up may appear recommending that you set the authentication method as LDAP. This recommended configuration should not be applied for AD, so you should clear (un-check) the `ranger.authentication.method` check box, then click **OK**.

The screenshot shows the 'Dependent Configurations' dialog box. It has a yellow header with the text: 'Based on your configuration changes, Ambari is recommending the following dependent configuration changes. Ambari will update all checked configuration changes to the Recommended Value. Uncheck any configuration to retain the Current Value.' Below this is a table with the following columns: Property, Service, Config Group, File Name, Current Value, and Recommended Value. The table contains one row: 'ranger.authentication.method' (Property), 'Ranger' (Service), 'Default' (Config Group), 'ranger-admin-site' (File Name), 'UNIX' (Current Value), and 'LDAP' (Recommended Value). The checkbox next to the 'Property' column for this row is checked and highlighted with a red square. At the bottom right of the dialog are 'Cancel' and 'OK' buttons.

Property	Service	Config Group	File Name	Current Value	Recommended Value
<input checked="" type="checkbox"/> ranger.authentication.method	Ranger	Default	ranger-admin-site	UNIX	LDAP

3.3. Complete the Ranger Installation

1. On the Review page, carefully review all of your settings and configurations. If everything looks good, click **Deploy** to install Ranger on the Ambari server.

Add Service Wizard

ADD SERVICE WIZARD

- Choose Services
- Assign Masters
- Assign Slaves and Clients
- Customize Services
- Configure Identities
- Review**
- Install, Start and Test
- Summary

Review

Please review the configuration before installation

Admin Name : admin
 Cluster Name : Thomas1
 Total Hosts : 3 (0 new)

Repositories:

- redhat5 (HDP-2.2):
http://public-repo-1.hortonworks.com/HDP/centos5/2.x/updates/2.2.6.0
- redhat5 (HDP-UTILS-1.1.0.20):
http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.20/repos/centos5
- redhat6 (HDP-2.2):
http://public-repo-1.hortonworks.com/HDP/centos6/2.x/updates/2.2.6.0
- redhat6 (HDP-UTILS-1.1.0.20):
http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.20/repos/centos6
- suse11 (HDP-2.2):
http://public-repo-1.hortonworks.com/HDP/suse11sp3/2.x/updates/2.2.6.0
- suse11 (HDP-UTILS-1.1.0.20):
http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.20/repos/suse11sp3
- ubuntu12 (HDP-2.2):
http://public-repo-1.hortonworks.com/HDP/ubuntu12/2.x/updates/2.2.6.3

← Back Print **Deploy** →

2. When you click **Deploy**, Ranger is installed on the specified host on your Ambari server. A progress bar displays the installation progress.

Add Service Wizard

ADD SERVICE WIZARD

- Choose Services
- Assign Masters
- Assign Slaves and Clients
- Customize Services
- Configure Identities
- Review
- Install, Start and Test**
- Summary

Install, Start and Test

Please wait while the selected services are installed and started.

24 % overall

Show: All (3) | In Progress (3) | Warning (0) | Success (0) | Fail (0)

Host	Status	Message
c6401.ambari.apache.org	6%	Installing Ranger Admin
c6402.ambari.apache.org	33%	Install complete (Waiting to start)
c6403.ambari.apache.org	33%	Install complete (Waiting to start)

3 of 3 hosts showing - Show All Show: 25 | 1 - 3 of 3

Next →

3. When the installation is complete, a Summary page displays the installation details. You may need to restart services for cluster components after installing Ranger.



Note

If the installation fails, you should complete the installation process, then reconfigure and reinstall Ranger.

3.4. Configuring Ranger for LDAP SSL

3.4.1. Import the LDAP Cert into the Default Java TrustStore

1. If you are using a CA signed certificate for your LDAP authentication, the certificate should already be included in the default Java trustStore located at `$JAVA_HOME/jre/lib/security/cacerts` on all of your nodes, or at least on the NameNode and Ranger Admin/Usersync nodes.
2. There is no need to manually restart Ranger or perform any keytool imports.
3. If necessary you can import the CA cert to `$JAVA_HOME/jre/lib/security/cacerts`. If you are using a self-signed cert you can use the keytool to import it into `$JAVA_HOME/jre/lib/security/cacerts`.

3.4.2. Alternative Option

You can also use the following method when the self-signed cert is not in `$JAVA_HOME/jre/lib/security/cacerts`.

For Ranger Usersync:

1. Edit `/usr/hdp/current/ranger-usersync/ranger-usersync-services.sh`.
2. Add java option `> -Djavax.net.ssl.trustStore=/<path to the cacert>`.

For Ranger Admin:

1. Edit `/usr/hdp/current/ranger-admin/ews/ranger-admin-services.sh`.
2. Add parameter `-Djavax.net.ssl.trustStore=/<path to the cacert>` to the Java call in the script.

3.5. Setting up Database Users Without Sharing DBA Credentials

If do not wish to provide system Database Administrator (DBA) account details to the Ambari Ranger installer, you can use the `dba_script.py` Python script to create Ranger DB database users without exposing DBA account information to the Ambari Ranger installer. You can then run the normal Ambari Ranger installation without specifying a DBA user name and password.

To create Ranger DB users using the `dba_script.py` script:

1. Download the Ranger rpm using the yum install command.

```
yum install ranger-admin
```

2. You should see one file named `dba_script.py` in the `/usr/hdp/current/ranger-admin` directory.
3. Get the script reviewed internally and verify that your DBA is authorized to run the script.
4. Execute the script by running the following command:

```
python dba_script.py
```

5. Pass all values required in the argument. These should include `db flavor`, `JDBC jar`, `db host`, `db name`, `db user`, and other parameters.
 - If you would prefer not to pass runtime arguments via the command prompt, you can update the `/usr/hdp/current/ranger-admin/install.properties` file and then run:

```
python dba_script.py -q
```

When you specify the `-q` option, the script will read all required information from the `install.properties` file

- You can use the `-d` option to run the script in "dry" mode. Running the script in dry mode causes the script to generate a database script.

```
python dba_script.py -d /tmp/generated-script.sql
```

Anyone can run the script, but it is recommended that the system DBA run the script in dry mode. In either case, the system DBA should review the generated script, but should only make minor adjustments to the script, for example, change the location of a particular database file. No major changes should be made that substantially alter the script – otherwise the Ranger install may fail.

The system DBA must then run the generated script.

6. Run the Ranger Ambari install procedure, but set **Setup Database and Database User** to **No** in the Ranger Admin section of the Customize Services screen.

3.6. Updating Ranger Admin Passwords

For the following users, if you update the passwords on the Ranger Configs page, you must also update the passwords on the Configs page of each Ambari component that has the Ranger plugin enabled. Individual Ambari component configurations are not automatically updated – the service restart will fail if you do not update these passwords on each component.

- Ranger Admin user – The credentials for this user are set in **Configs > Advanced ranger-env** in the fields labeled **admin_username** (default value: `admin`) and **admin_password** (default value: `admin`).
- Admin user used by Ambari to create repo/policies – The user name for this user is set in **Configs > Admin Settings** in the field labeled **Ranger Admin username for Ambari**

(default value: amb_ranger_admin). The password for this user is set in the field labeled **Ranger Admin user's password for Ambari**. This password is specified during the Ranger installation.

The following image shows the location of these settings on the Ranger Configs page:

The screenshot displays the Ambari Ranger configuration interface. On the left is a navigation sidebar with various services listed, including HDFS, MapReduce2, YARN, Tez, Hive, HBase, Pig, Sqoop, Oozie, ZooKeeper, Falcon, Storm, Flume, Accumulo, Ambari Metrics, Atlas, Kafka, Knox, Mahout, Ranger, Slider, SmartSense, and Spark. The 'Ranger' service is selected and highlighted in red.

The main content area shows the 'Ranger Configs' page for the 'Default (1)' group. At the top, there are tabs for 'Summary', 'Configs', 'Quick Links', and 'Service Actions'. Below this is a 'Manage Config Groups' section with a 'Filter...' input and a dropdown menu.

The 'Admin Settings' section is expanded, showing the following configuration items:

- Ranger Admin host: c6403.ambari.apache.org
- Ranger Admin username for Ambari: amb_ranger_admin
- Ranger Admin user's password for Ambari: (masked)
- Location of Sql Connector Jar: /usr/share/java/mysql-connector-java.jar

The 'Advanced ranger-env' section is also expanded, showing the following configuration items:

- Ranger Group: ranger
- Ranger User: ranger
- admin_password: (masked)
- admin_username: admin
- ranger_admin_log_dir: /var/log/ranger/admin
- ranger_pid_dir: /var/run/ranger
- ranger_usersync_log_dir: /var/log/ranger/usersync

A purple box highlights the 'Ranger Admin username for Ambari' and 'Ranger Admin user's password for Ambari' fields. A red box highlights the 'admin_password' and 'admin_username' fields in the 'Advanced ranger-env' section. A red label 'Ranger "admin" user details' points to the red box.

4. Using Apache Solr for Ranger Audits

Apache Solr is an open-source enterprise search platform. Apache Ranger can use Apache Solr to store audit logs, and Solr can also provide a search capability of the audit logs through the Ranger Admin UI.



Important

Solr must be installed and configured before installing RangerAdmin or any of the Ranger component plugins.

It is recommended that Ranger audits be written to both Solr and HDFS. Audits to Solr are primarily used to enable search queries from the Ranger Admin UI. HDFS is a long-term destination for audits – audits stored in HDFS can be exported to any SIEM system, or to another audit store.

Configuration Options

- Solr Standalone – Solr Standalone is only recommended for testing and evaluation. Solr Standalone is a single instance of Solr that does not require ZooKeeper.
- SolrCloud – This is the recommended configuration for Ranger. [SolrCloud](#) is a scalable architecture that can run as single node or as a multi-node cluster. It includes features such as replication and sharding, which are useful for high availability (HA) and scalability. With SolrCloud, you need to plan the deployment based on the cluster size.

The following sections describe how to install and configure Apache Solr for Ranger Audits:

- [Prerequisites \[40\]](#)
- [Installing Solr \[41\]](#)
- [Configuring Solr Standalone \[41\]](#)
- [Configuring SolrCloud \[42\]](#)

4.1. Prerequisites

Solr Prerequisites

- Ranger supports Apache Solr 5.2 or higher.
- Apache Solr requires the Java Runtime Environment (JRE) version 1.7 or higher.
- 1 TB free space in the volume where Solr will store the index data.
- 32 GB RAM.

SolrCloud Prerequisites

- SolrCloud supports replication and sharding. It is highly recommended that you use SolrCloud with at least two Solr nodes running on different servers with replication enabled.

- SolrCloud requires Apache ZooKeeper.

4.2. Installing Solr

Use the followingn command to install Solr:

```
yum install lucidworks-hdpsearch
```

The HDP Search installer installs Solr in the `/opt/lucidworks-hdpsearch/solr` directory.

4.3. Configuring Solr Standalone

Use the following procedure to configure Solr Standalone.

1. Download the `solr_for_audit_setup_v3` file to the `/usr/local/` directory:

```
wget https://issues.apache.org/jira/secure/attachment/12761323/solr_for_audit_setup_v3.tgz -O /usr/local/solr_for_audit_setup_v3.tgz
```

2. Use the following commands to switch to the `/usr/local/` directory and extract the `solr_for_audit_setup_v3` file.

```
cd /usr/local
tar xvf solr_for_audit_setup_v3.tgz
```

The contents of the `.tgz` file will be extracted into a `/usr/local/solr_for_audit_setup` directory.

3. Use the following command to switch to the `/usr/local/solr_for_audit_setup` directory.

```
cd /usr/local/solr_for_audit_setup
```

4. Use the following command to open the `install.properties` file in the `vi` text editor.

```
vi install.properties
```

Set the following property values, then save the changes to the `install.properties` file.

Table 4.1. Solr install.properties Values

Property Name	Value	Description
JAVA_HOME	<path_to_jdk>, for example: <code>/usr/jdk64/jdk1.8.0_60</code>	Provide the path to the JDK install folder. For Hadoop, you can check <code>/etc/hadoop/conf/hadoop-env.sh</code> for the value of <code>JAVA_HOME</code> . As noted previously, Solr only supports JDK 1.7 and higher.
SOLR_USER	<code>solr</code>	The Linux user used to run Solr.
SOLR_INSTALL_FOLDER	<code>/opt/lucidworks-hdpsearch/solr</code>	The Solr installation directory.
SOLR_RANGER_HOME	<code>/opt/lucidworks-hdpsearch/solr/ranger_audit_server</code>	The location where the Ranger-related configuration and schema files will be copied.

Property Name	Value	Description
SOLR_RANGER_PORT	6083	The Solr port for Ranger.
SOLR_DEPLOYMENT	standalone	The deployment type.
SOLR_RANGER_DATA_FOLDER	/opt/lucidworks-hdpsearch/solr/ranger_audit_server/data	The folder where the index data will be stored. The volume for this folder should have at least 1 TB free space for the index data, and should be backed up regularly.
SOLR_LOG_FOLDER	/var/log/solr/ranger_audits	The folder for the Solr log files.
SOLR_MAX_MEM	2g	The memory allocation for Solr.

- Use the following command to run the Solr for Ranger setup script.

```
./setup.sh
```

- To start Solr, log in as the `solr` or `root` user and run the following command.

```
/opt/lucidworks-hdpsearch/solr/ranger_audit_server/scripts/start_solr.sh
```

When Solr starts, a confirmation message appears.

```
Started Solr server on port 6083 (pid=). Happy searching!
```

- You can use a web browser to open the Solr Admin Console at the following address:

```
http:<solr_host>:6083/solr
```



Note

You can use the following command to stop Solr:

```
/opt/lucidworks-hdpsearch/solr/ranger_audit_server/scripts/stop_solr.sh
```

4.4. Configuring SolrCloud

Use the following procedure to configure SolrCloud.

- Download the `solr_for_audit_setup_v3` file to the `/usr/local/` directory:

```
wget https://issues.apache.org/jira/secure/attachment/12761323/solr_for_audit_setup_v3.tgz -O /usr/local/solr_for_audit_setup_v3.tgz
```

- Use the following commands to switch to the `/usr/local/` directory and extract the `solr_for_audit_setup_v3` file.

```
cd /usr/local
tar xvf solr_for_audit_setup_v3.tgz
```

The contents of the `.tgz` file will be extracted into a `/usr/local/solr_for_audit_setup` directory.

- Use the following command to switch to the `/usr/local/solr_for_audit_setup` directory.

```
cd /usr/local/solr_for_audit_setup
```

- Use the following command to open the `install.properties` file in the vi text editor.

```
vi install.properties
```

Set the following property values, then save the changes to the `install.properties` file.

Table 4.2. Solr install.properties Values

Property Name	Value	Description
JAVA_HOME	<path_to_jdk>, for example: <code>/usr/jdk64/jdk1.8.0_40</code>	Provide the path to the JDK install folder. For Hadoop, you can check <code>/etc/hadoop/conf/hadoop-env.sh</code> for the value of <code>JAVA_HOME</code> . As noted previously, Solr only supports JDK 1.7 and higher.
SOLR_USER	<code>solr</code>	The Linux user used to run Solr.
SOLR_INSTALL_FOLDER	<code>/opt/lucidworks-hdpsearch/solr</code>	The Solr installation directory.
SOLR_RANGER_HOME	<code>/opt/lucidworks-hdpsearch/solr/ranger_audit_server</code>	The location where the Ranger-related configuration and schema files will be copied.
SOLR_RANGER_PORT	<code>6083</code>	The Solr port for Ranger.
SOLR_DEPLOYMENT	<code>solrcloud</code>	The deployment type.
SOLR_ZK	<ZooKeeper_host>:2181/ <code>ranger_audits</code>	The Solr ZooKeeper host and port. It is recommended to provide a sub-folder to create the Ranger Audit related configurations so you can also use ZooKeeper for other Solr instances. Due to a Solr bug, if you are using a path (sub-folder), you can only specify one ZooKeeper host.
SOLR_SHARDS	<code>1</code>	If you want to distribute your audit logs, you can use multiple shards. Make sure the number of shards is equal or less than the number of Solr nodes you will be running.
SOLR_REPLICATION	<code>1</code>	It is highly recommend that you set up at least two nodes and replicate the indexes. This gives redundancy to index data, and also provides load balancing of Solr queries.
SOLR_LOG_FOLDER	<code>/var/log/solr/ranger_audits</code>	The folder for the Solr log files.
SOLR_MAX_MEM	<code>2g</code>	The memory allocation for Solr.

- Use the following command to run the set up script.

```
./setup.sh
```

- Run the following command **only once** from any node. This command adds the Ranger Audit configuration (including `schema.xml`) to ZooKeeper.

```
/opt/lucidworks-hdpsearch/solr/ranger_audit_server/scripts/add_ranger_audits_conf_to_zk.sh
```

- Log in as the `solr` or `root` user and run the following command to start Solr on each node.

```
/opt/lucidworks-hdpsearch/solr/ranger_audit_server/scripts/start_solr.sh
```

When Solr starts, a confirmation message appears.

```
Started Solr server on port 6083 (pid=). Happy searching!
```

8. Run the following command **only once** from any node. This command creates the Ranger Audit collection.

```
/opt/lucidworks-hdpsearch/solr/ranger_audit_server/scripts/  
create_ranger_audits_collection.sh
```

9. You can use a web browser to open the Solr Admin Console at the following address:

```
http:<solr_host>:6083/solr
```



Note

You can use the following command to stop Solr:

```
/opt/lucidworks-hdpsearch/solr/ranger_audit_server/scripts/  
stop_solr.sh
```

5. Ranger Plug ins Overview

Ranger plugins can be enabled for several HDP services. This section describes how to enable each of these plugins. For performance reasons, it is recommended that you store audits in Solr and HDFS, and not in a database.

If you are using a Kerberos-enabled cluster, there are a number of additional steps you must follow to ensure that you can use the Ranger plugins on a Kerberos cluster.

The following Ranger plugins are available:

- [HDFS \[45\]](#)
- [Hive \[49\]](#)
- [HBase \[53\]](#)
- [Kafka \[56\]](#)
- [Knox \[60\]](#)
- [YARN \[63\]](#)
- [Storm \[67\]](#)

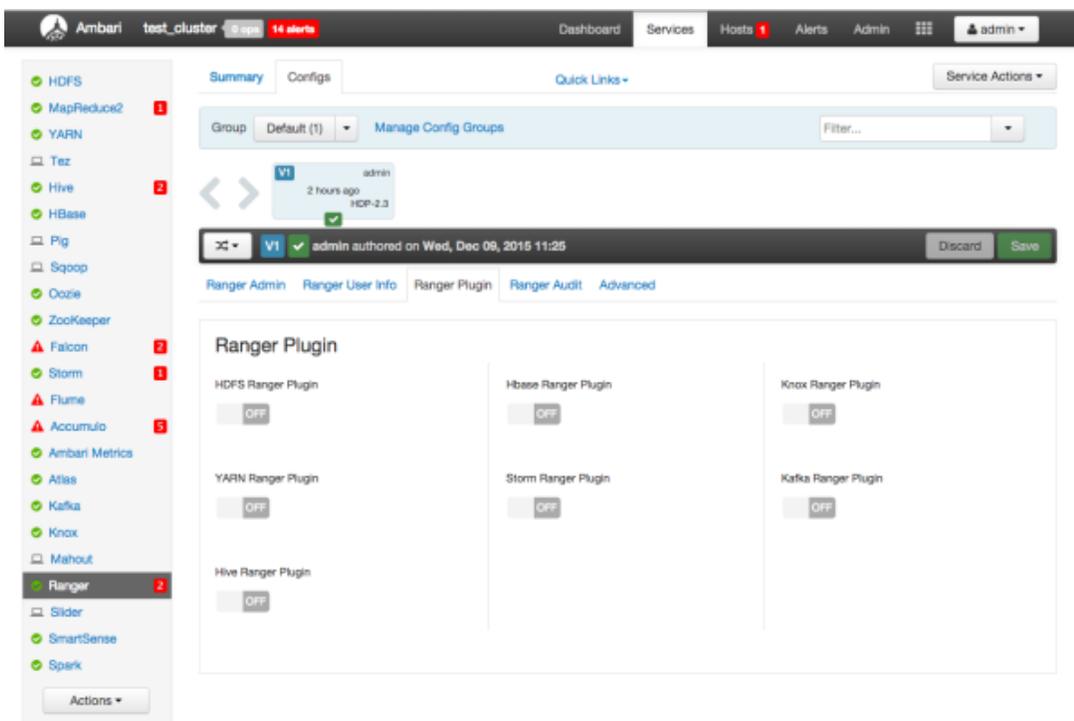
You can save Ranger audits to HDFS or Solr:

- [Manually Updating HDFS Audit Settings \[71\]](#)
- [Manually Updating Solr Audit Settings \[72\]](#)

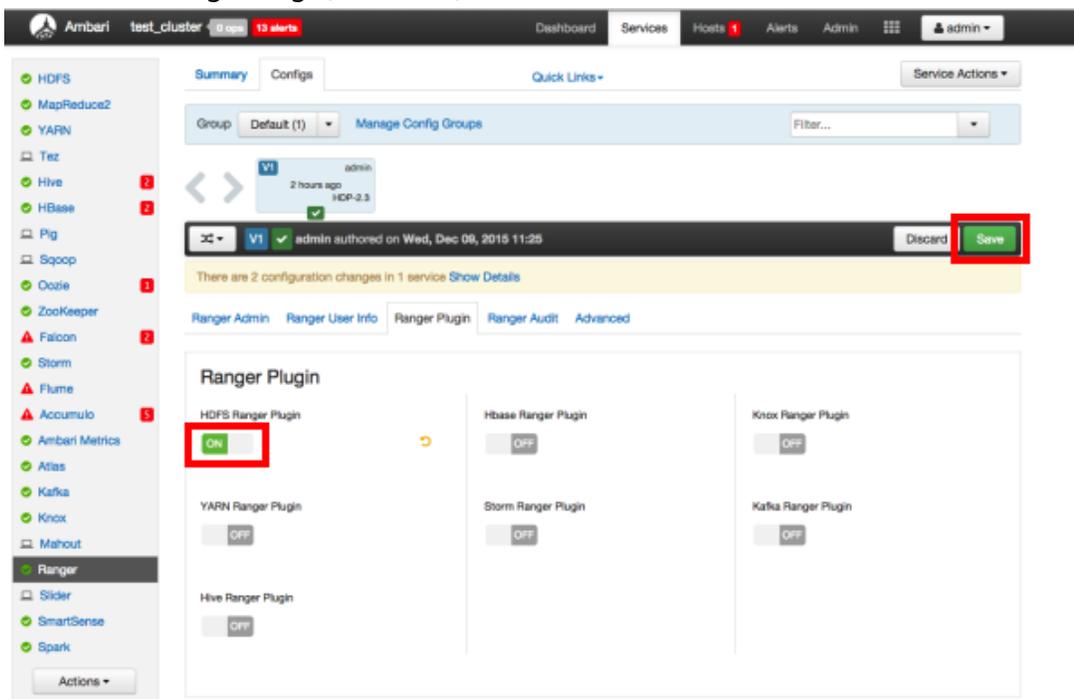
5.1. HDFS

Use the following steps to enable the Ranger HDFS plugin.

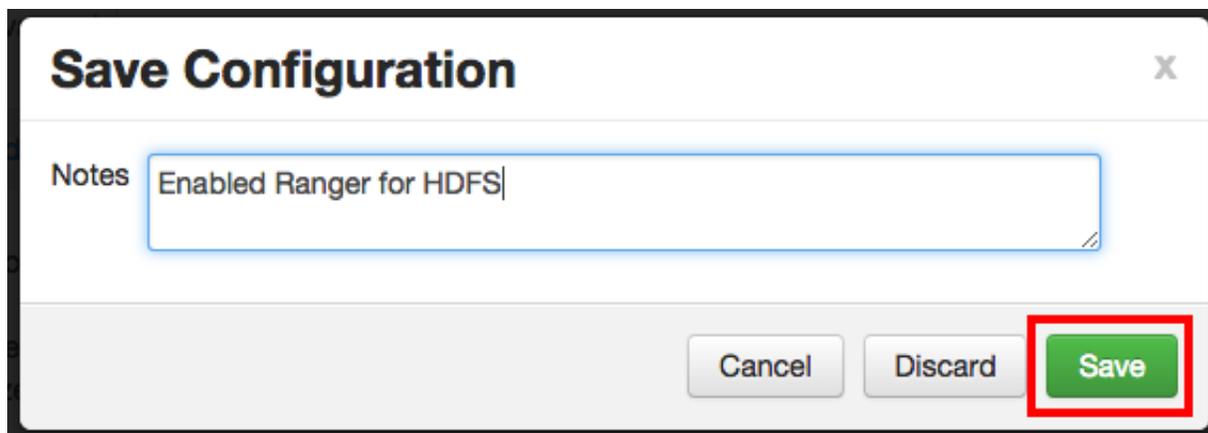
1. On the Ranger Configs page, select the **Ranger Plugin** tab.



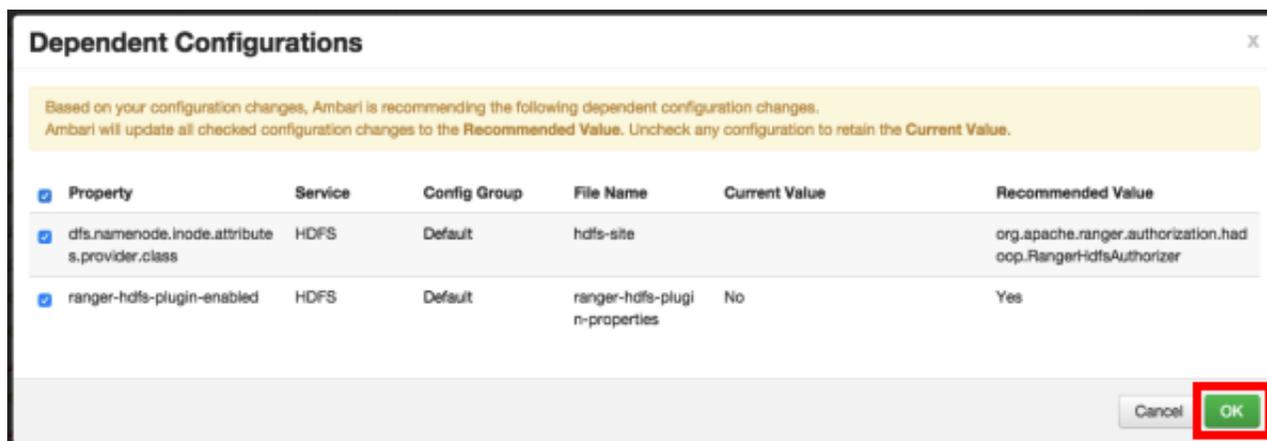
2. Under HDFS Ranger Plugin, select **On**, then click **Save** in the black menu bar.



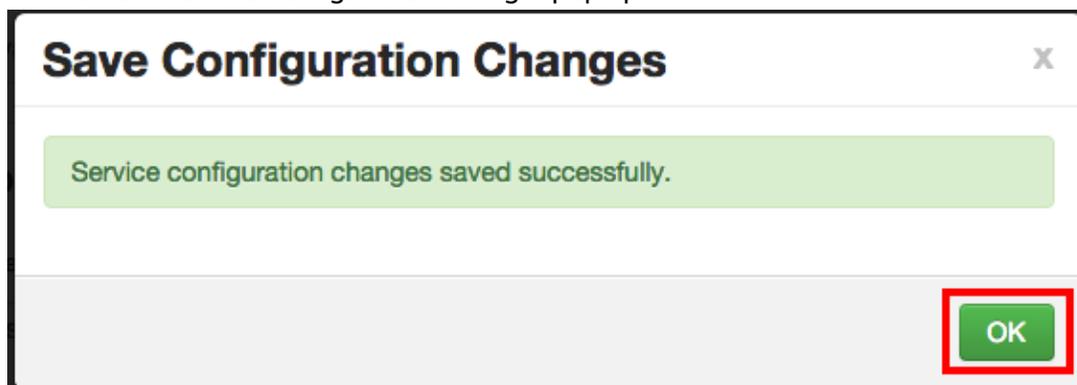
3. A Save Configuration pop-up appears. Type in a note describing the changes you just made, then click **Save**.



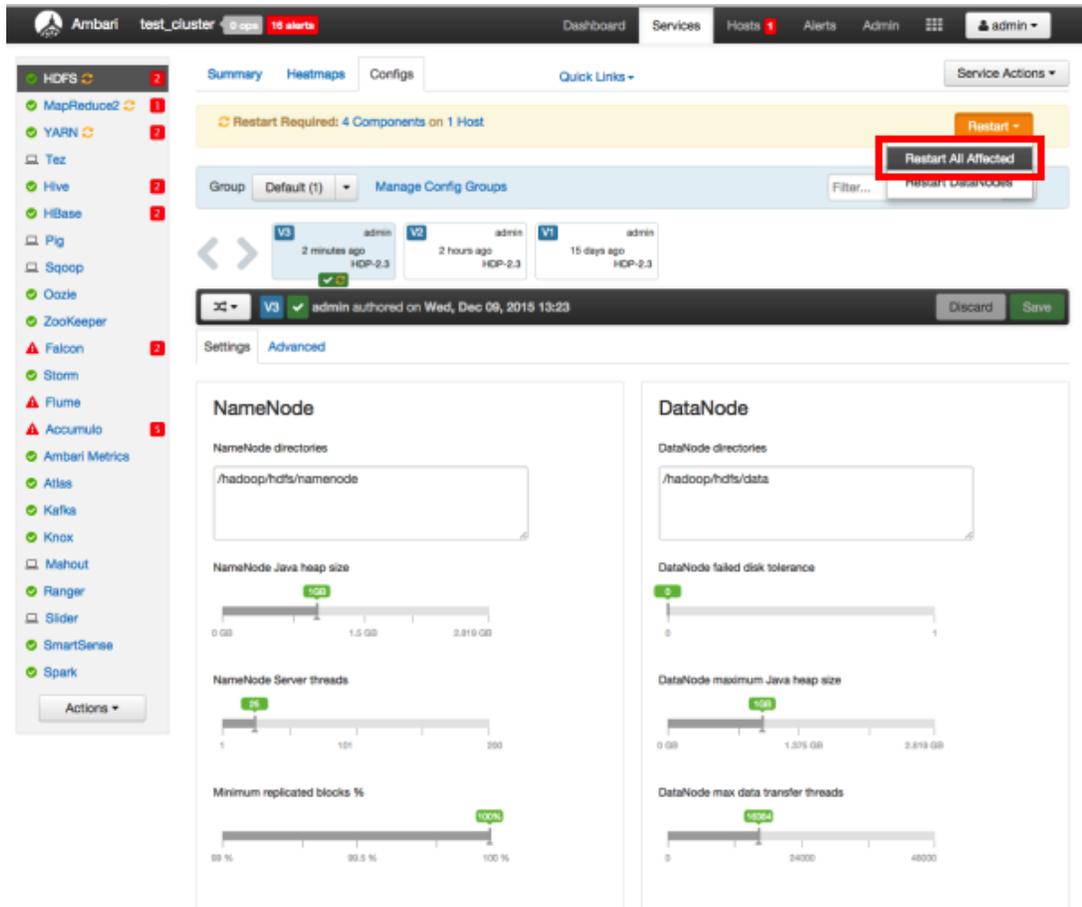
4. A Dependent Configuration pop-up appears. Click **OK** to confirm the configuration updates.



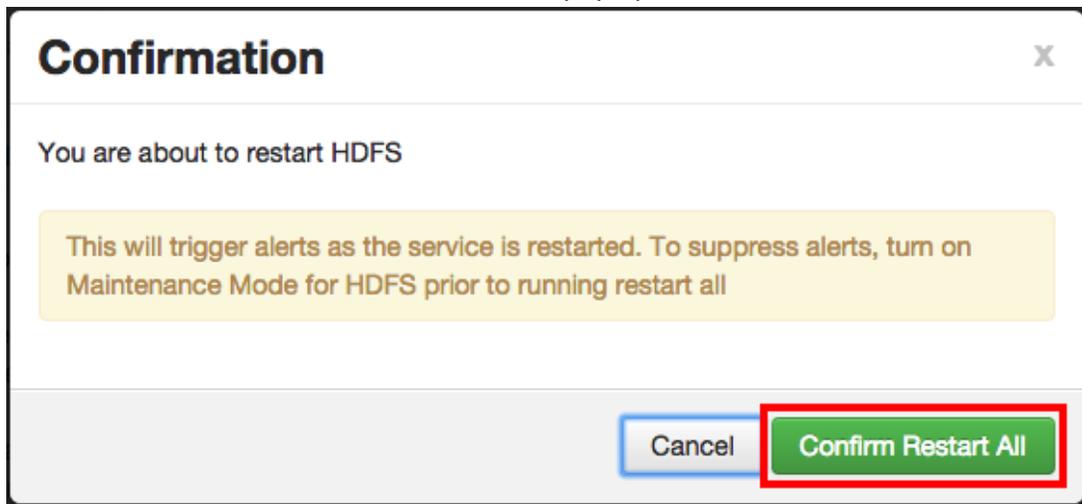
5. Click **OK** on the Save Configuration Changes pop-up.



6. Select **HDFS** in the navigation menu, then select **Restart > Restart All Affected** to restart the HDFS service and load the new configuration.



7. Click **Confirm Restart All** on the confirmation pop-up to confirm the HDFS restart.



8. After HDFS has restarted, the Ranger plugin for HDFS is enabled.



Note

In order to access HDFS folders in previous versions of HDP, access permissions also had to be granted in Ranger to the applicable parent folders. As of HDP-2.3, it is no longer required to grant access permissions to the parent folder.

For example, for the folder path `.. /customer/data/marketing`:

- In previous versions, to grant access to the `/customer/data/marketing` folder, you were required to grant Execute permission in Ranger for both the `/customer` and `/customer/data` folders, along with a Read or Write permission for the `/customer/data/marketing` folder.
- As of HDP-2.3, it is no longer necessary to grant Execute permission to the parent folders.

For more details, see [RANGER-357](#).

5.2. Hive

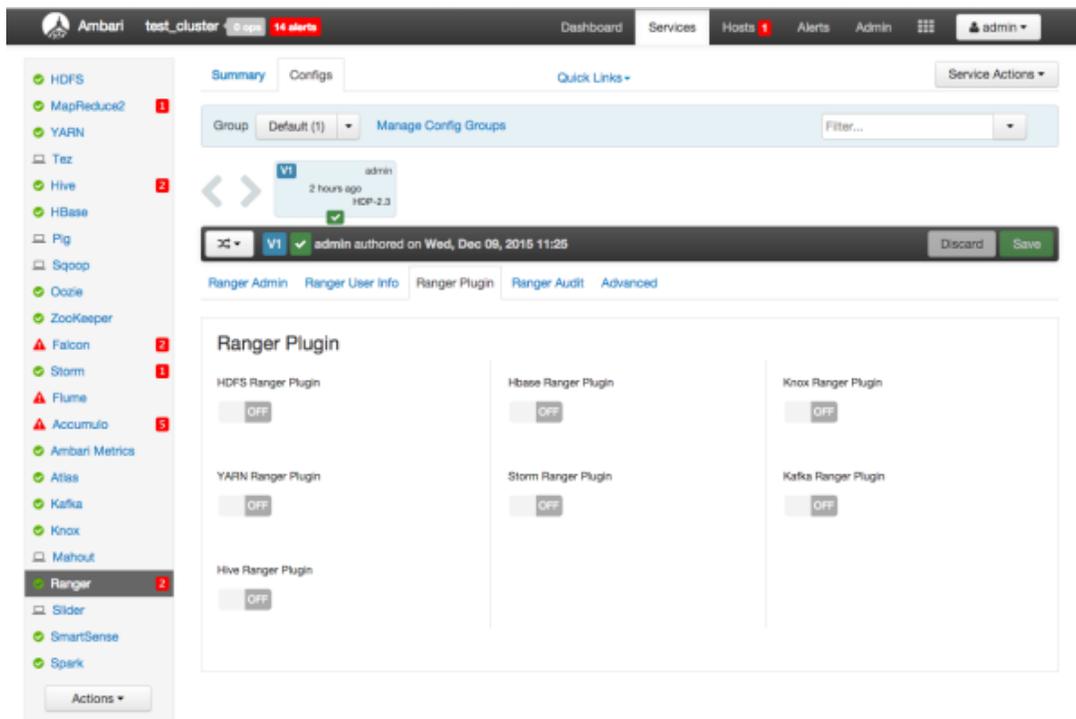


Important

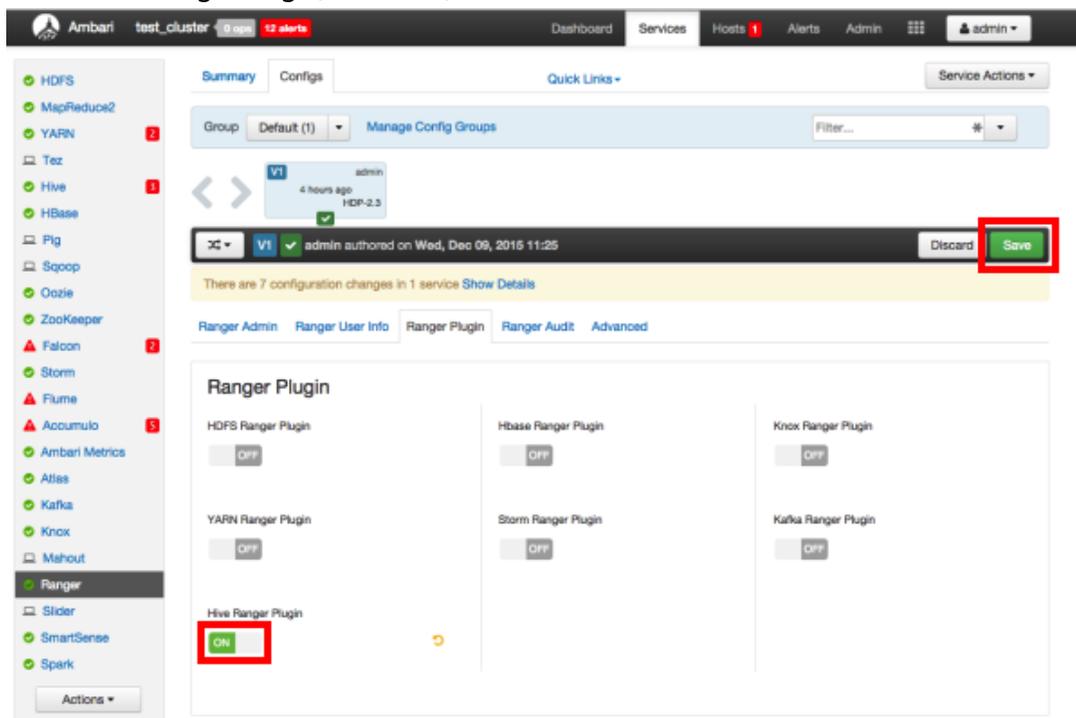
You should not use the Hive CLI after enabling the Ranger Hive plugin. The Hive CLI is not supported in HDP-2.2.0 and higher versions, and may break the install or lead to other unpredictable behavior. Instead, you should use the [HiveServer2 Beeline CLI](#).

Use the following steps to enable the Ranger Hive plugin.

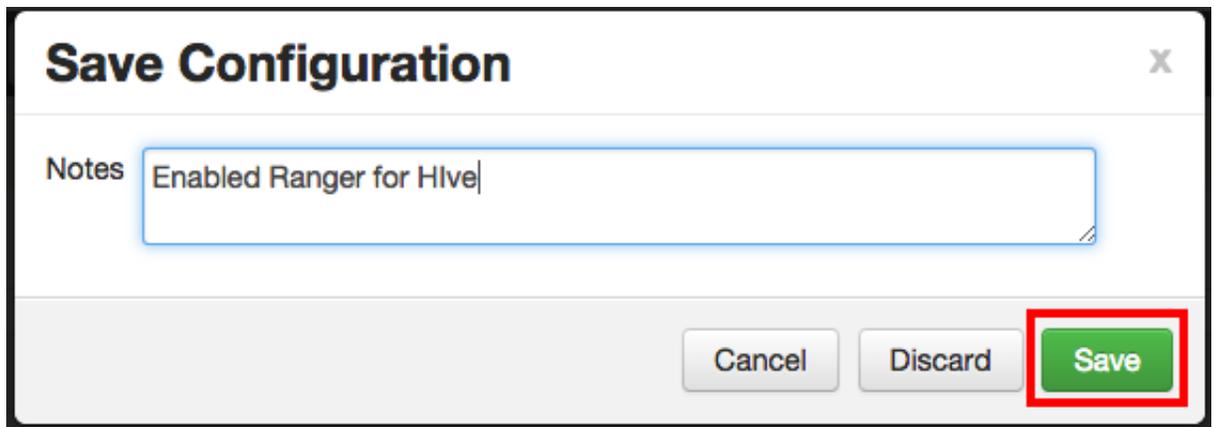
1. On the Ranger Configs page, select the **Ranger Plugin** tab.



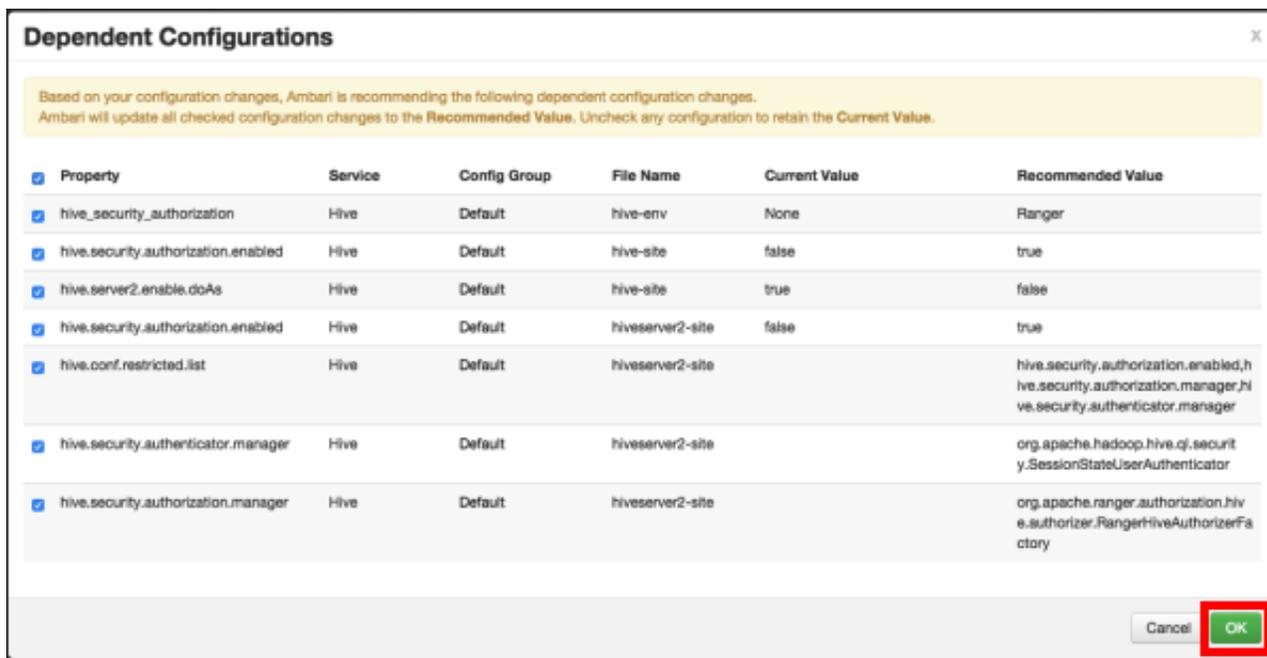
2. Under Hive Ranger Plugin, select **On**, then click **Save** in the black menu bar.



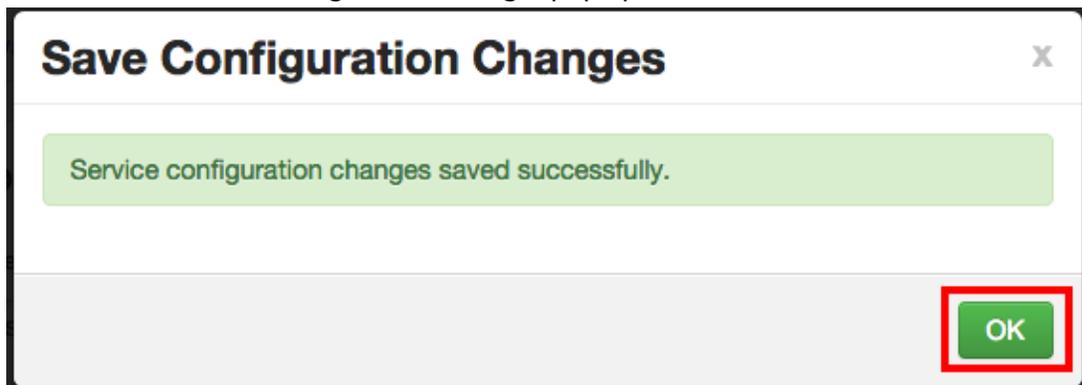
3. A Save Configuration pop-up appears. Type in a note describing the changes you just made, then click **Save**.



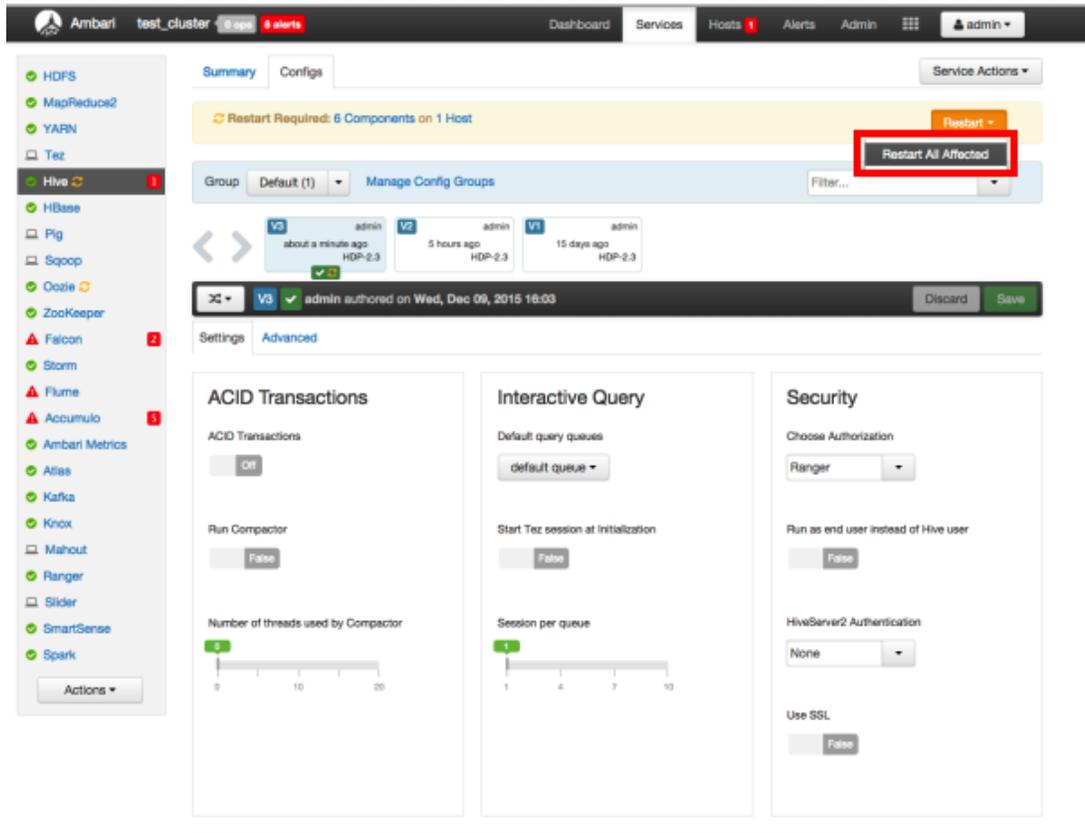
4. A Dependent Configuration pop-up appears. Click **OK** to confirm the configuration updates.



5. Click **OK** on the Save Configuration Changes pop-up.

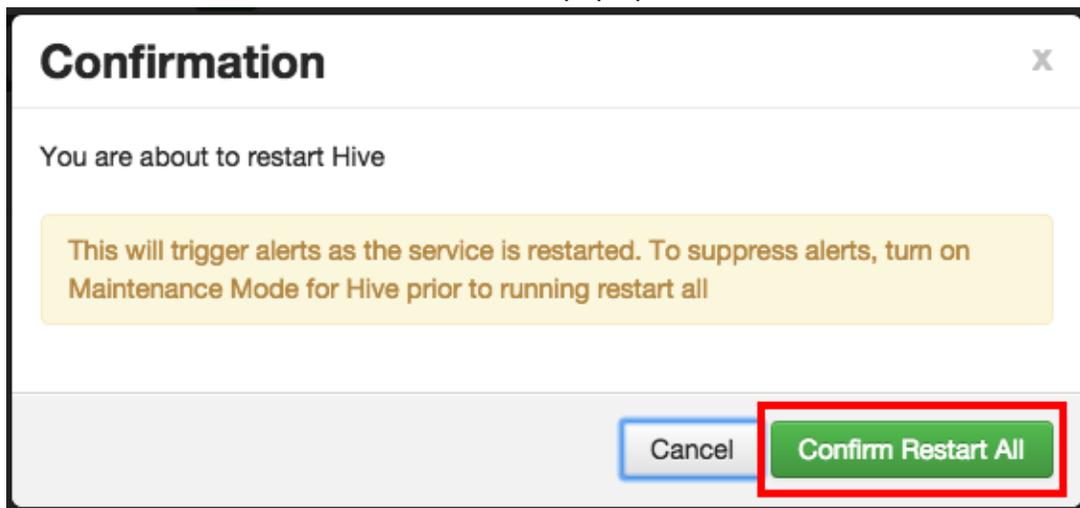


6. Select **Hive** in the navigation menu, then select **Restart > Restart All Affected** to restart the Hive service and load the new configuration.



The screenshot shows the Ambari web interface for the Hive service. The top navigation bar includes 'Dashboard', 'Services', 'Hosts', 'Alerts', and 'Admin'. The left sidebar lists various services, with 'Hive' selected and highlighted in blue. The main content area shows the 'Summary' and 'Configs' tabs. A yellow banner at the top indicates 'Restart Required: 6 Components on 1 Host'. Below this, there is a 'Restart' button and a 'Restart All Affected' button, the latter of which is highlighted with a red rectangular box. The interface also displays a list of hosts (V3, V2, V1) and their configurations, along with a 'Settings' section for 'Advanced' configuration options like 'ACID Transactions', 'Interactive Query', and 'Security'.

7. Click **Confirm Restart All** on the confirmation pop-up to confirm the Hive restart.



The screenshot shows a 'Confirmation' dialog box with a close button (X) in the top right corner. The main text reads 'You are about to restart Hive'. Below this is a yellow warning box containing the text: 'This will trigger alerts as the service is restarted. To suppress alerts, turn on Maintenance Mode for Hive prior to running restart all'. At the bottom of the dialog, there are two buttons: 'Cancel' and 'Confirm Restart All'. The 'Confirm Restart All' button is highlighted with a red rectangular box.

8. After Hive restarts, the Ranger plugin for Hive will be enabled.

5.3. HBase



Note

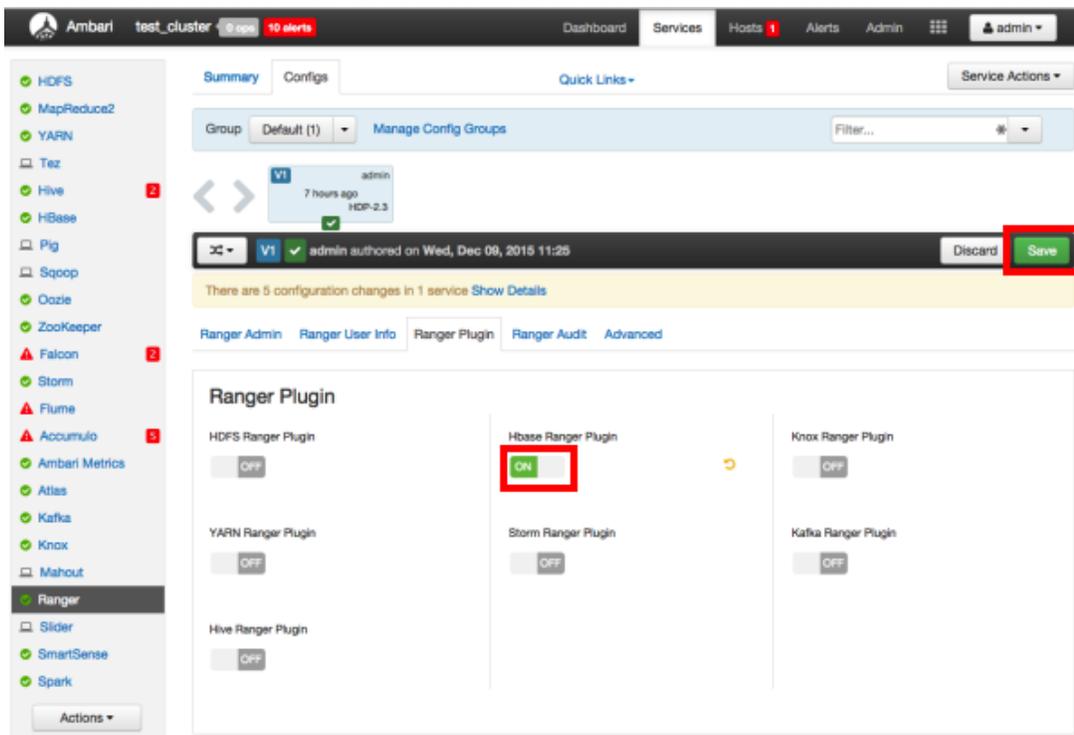
When HBase is configured with Ranger, and specifically XASecure Authorizer, you may only grant and revoke privileges.

Use the following steps to enable the Ranger HBase plugin.

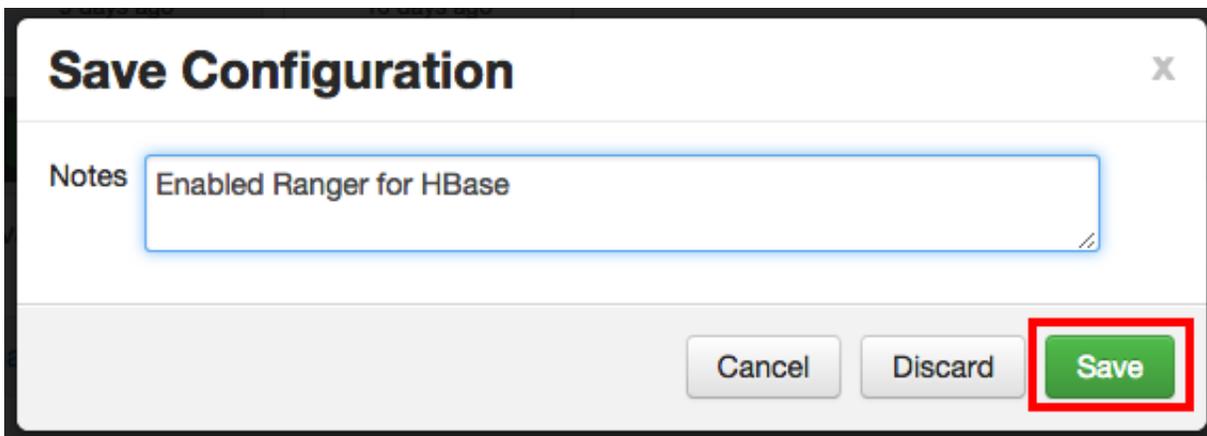
1. On the Ranger Configs page, select the **Ranger Plugin** tab.

The screenshot shows the Ambari Ranger Admin interface. The top navigation bar includes 'Ambari test_cluster', 'Dashboard', 'Services', 'Hosts', 'Alerts', 'Admin', and a user profile 'admin'. The main content area is titled 'Ranger Admin' and has tabs for 'Ranger Admin', 'Ranger User Info', 'Ranger Plugin', 'Ranger Audit', and 'Advanced'. The 'Ranger Plugin' tab is selected. The page displays a grid of toggle switches for various plugins: HDFS Ranger Plugin, Hbase Ranger Plugin, Knox Ranger Plugin, YARN Ranger Plugin, Storm Ranger Plugin, and Kafka Ranger Plugin. All switches are currently set to 'OFF'. A black menu bar at the top of the configuration area contains 'Discard' and 'Save' buttons. The left sidebar shows a list of services including HDFS, MapReduce2, YARN, Tez, Hive, HBase, Pig, Sqoop, Dozie, ZooKeeper, Falcon, Storm, Flume, Accumulo, Ambari Metrics, Atlas, Kafka, Knox, Mahout, Ranger, Slider, SmartSense, and Spark. The Ranger service is highlighted in the sidebar.

2. Under HBase Ranger Plugin, select **On**, then click **Save** in the black menu bar.



- 3. A Save Configuration pop-up appears. Type in a note describing the changes you just made, then click Save.



- 4. A Dependent Configuration pop-up appears. Click OK to confirm the configuration updates.

Dependent Configurations

Based on your configuration changes, Ambari is recommending the following dependent configuration changes.
Ambari will update all checked configuration changes to the Recommended Value. Uncheck any configuration to retain the Current Value.

<input checked="" type="checkbox"/>	Property	Service	Config Group	File Name	Current Value	Recommended Value
<input checked="" type="checkbox"/>	hbase.security.authorization	HBase	Default	hbase-site	false	true
<input checked="" type="checkbox"/>	hbase.coprocessor.regionserver.classes	HBase	Default	hbase-site		org.apache.ranger.authorization.hbase.RangerAuthorizationCoproces
<input checked="" type="checkbox"/>	hbase.coprocessor.master.classes	HBase	Default	hbase-site		org.apache.ranger.authorization.hbase.RangerAuthorizationCoproces
<input checked="" type="checkbox"/>	hbase.coprocessor.region.classes	HBase	Default	hbase-site	org.apache.hadoop.hbase.security.a	org.apache.hadoop.hbase.security.a
<input checked="" type="checkbox"/>	ranger-hbase-plugin-enabled	HBase	Default	ranger-hbase-plugin-pr	No	Yes

Cancel **OK**

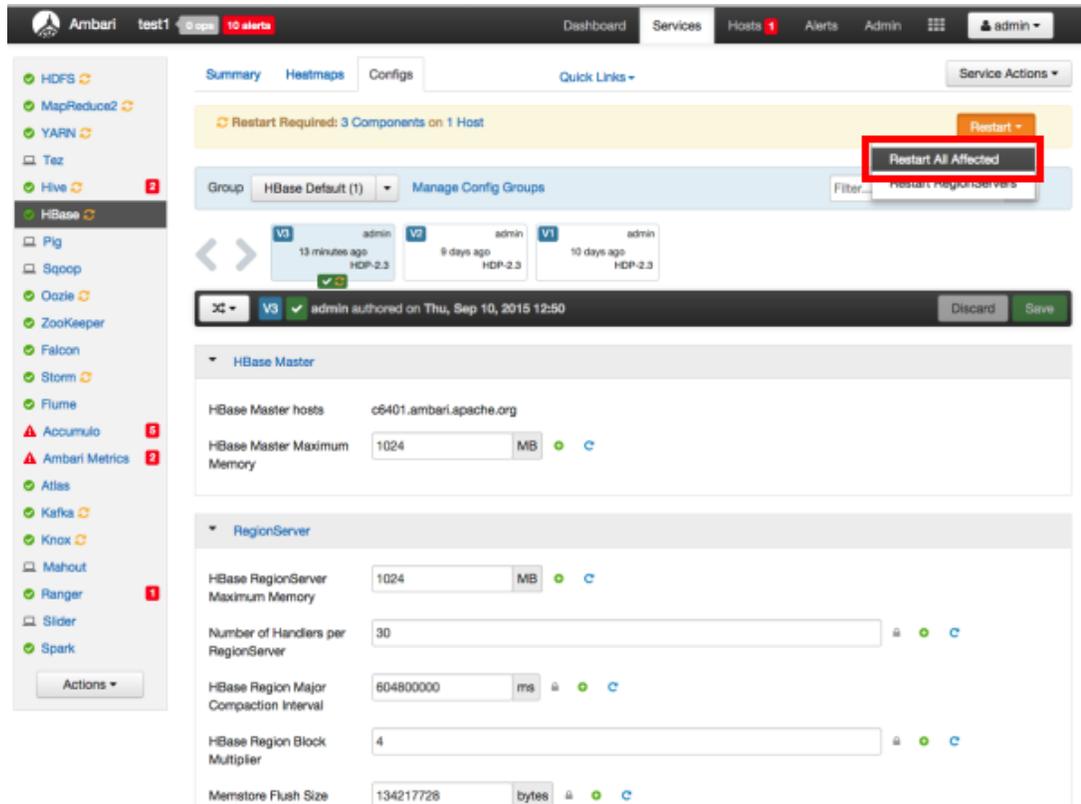
5. Click **OK** on the Save Configuration Changes pop-up.

Save Configuration Changes

Service configuration changes saved successfully.

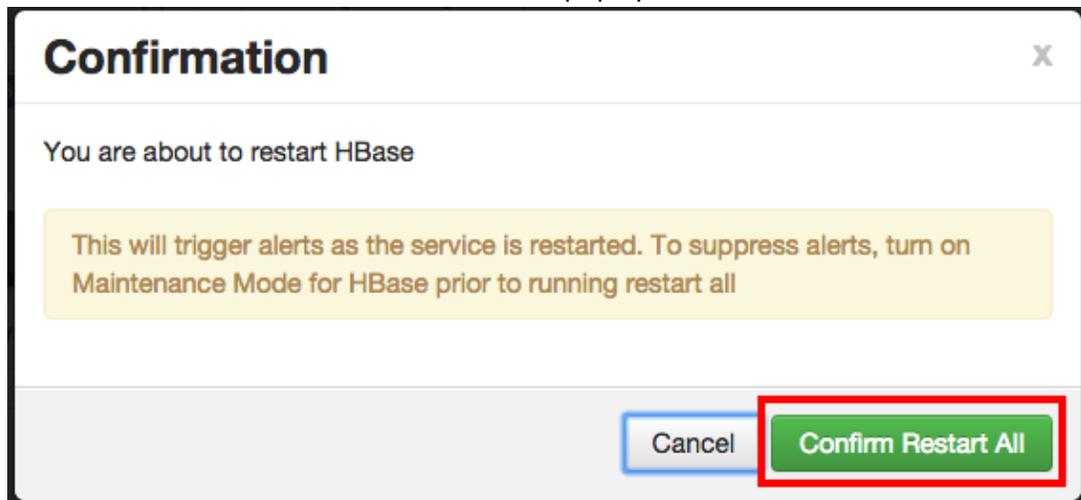
OK

6. Select **HBase** in the navigation menu, then select **Restart > Restart All Affected** to restart the HBase service and load the new configuration.



The screenshot shows the Ambari interface for the HBase service. At the top, there's a navigation bar with 'Dashboard', 'Services', 'Hosts', 'Alerts', and 'Admin'. The main content area has tabs for 'Summary', 'Heatmaps', and 'Configs'. A yellow alert banner at the top indicates 'Restart Required: 3 Components on 1 Host'. Below this, there's a 'Restart' button and a 'Restart All Affected' button, the latter being highlighted with a red box. The configuration section below shows settings for 'HBase Master' and 'RegionServer', including 'Maximum Memory' and 'Number of Handlers per RegionServer'.

7. Click **Confirm Restart All** on the confirmation pop-up to confirm the HBase restart.



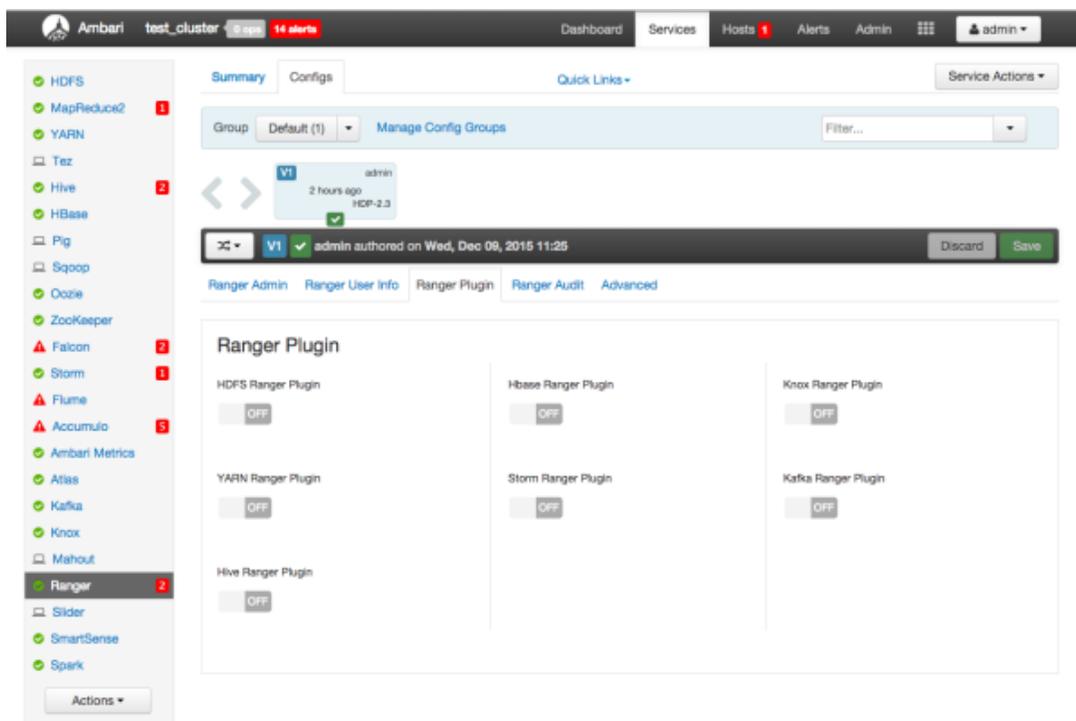
The screenshot shows a 'Confirmation' dialog box. The title is 'Confirmation'. The main text says 'You are about to restart HBase'. Below this is a yellow warning box with the text: 'This will trigger alerts as the service is restarted. To suppress alerts, turn on Maintenance Mode for HBase prior to running restart all'. At the bottom of the dialog, there are two buttons: 'Cancel' and 'Confirm Restart All', with the latter being highlighted by a red box.

8. After HBase restarts, the Ranger plugin for HBase will be enabled.

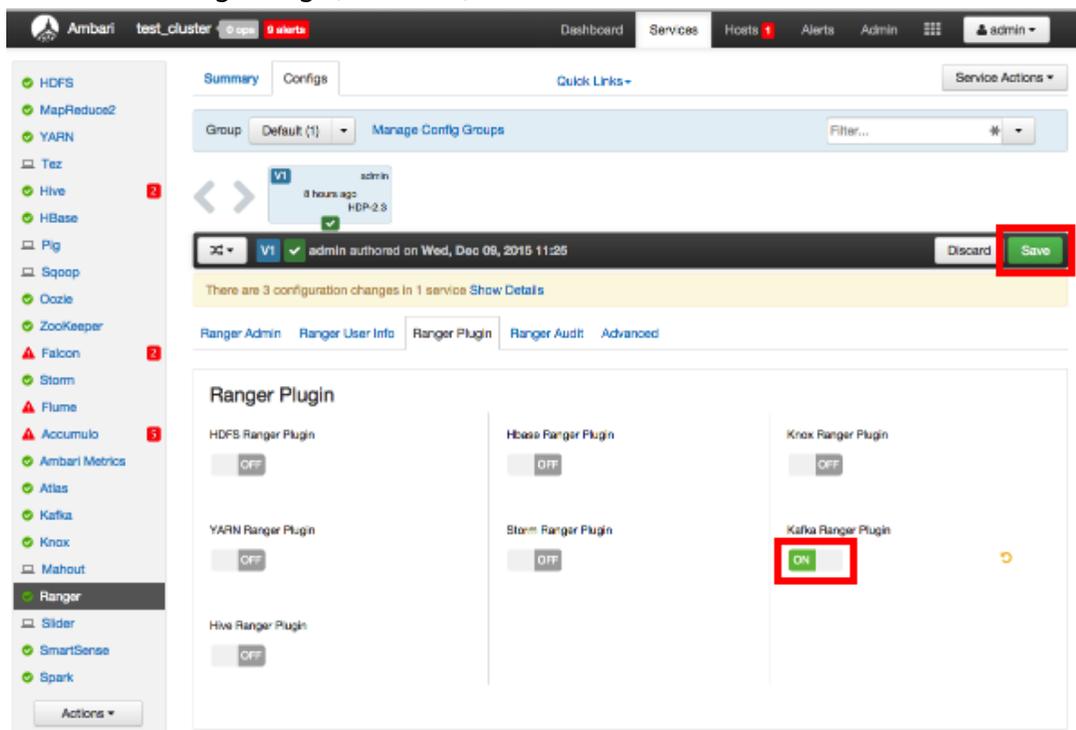
5.4. Kafka

Use the following steps to enable the Ranger Kafka plugin.

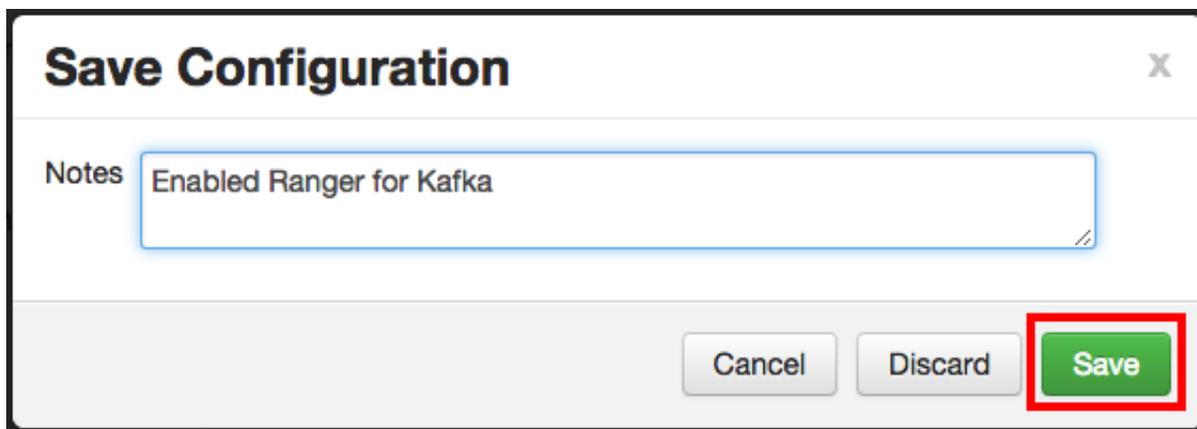
1. On the Ranger Configs page, select the **Ranger Plugin** tab.



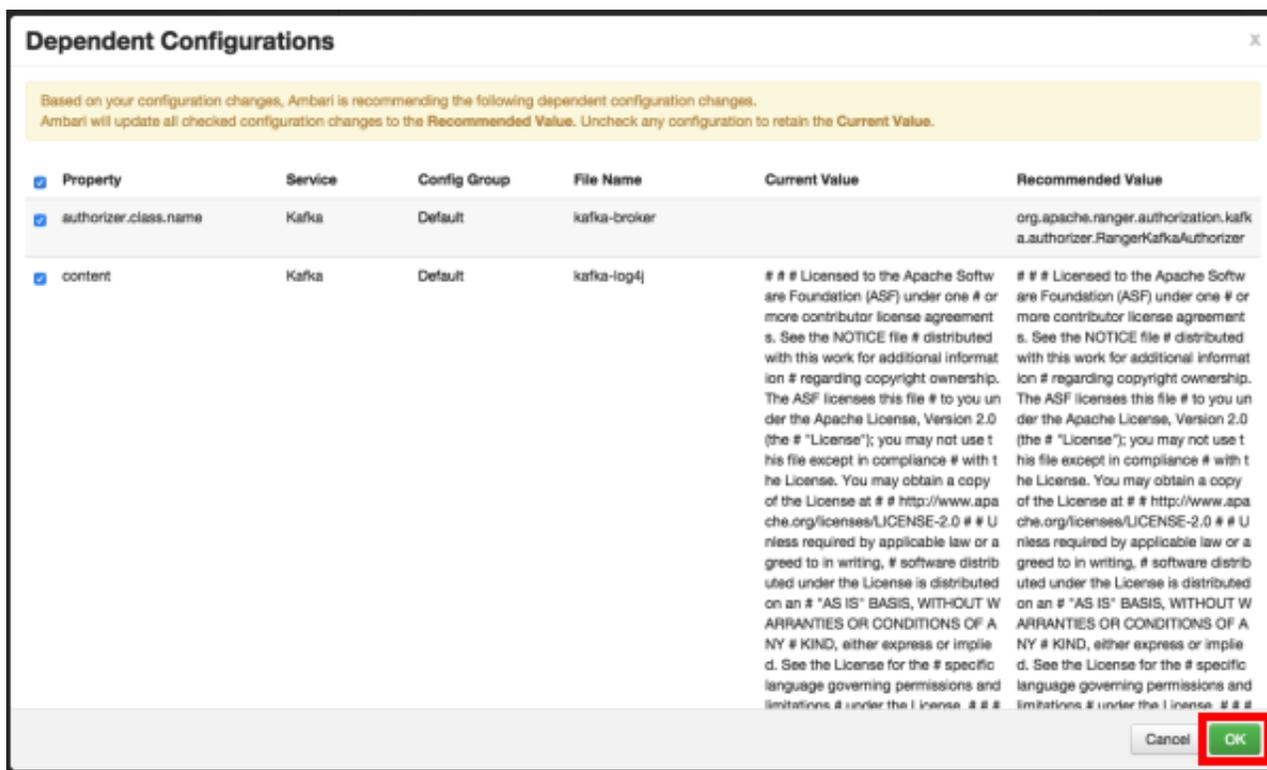
2. Under Kafka Ranger Plugin, select On, then click Save in the black menu bar.



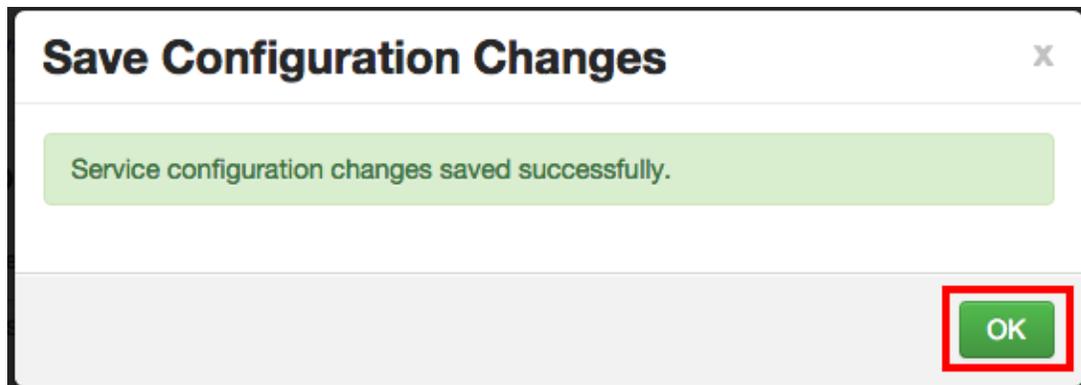
3. A Save Configuration pop-up appears. Type in a note describing the changes you just made, then click Save.



4. A Dependent Configuration pop-up appears. Click **OK** to confirm the configuration updates.

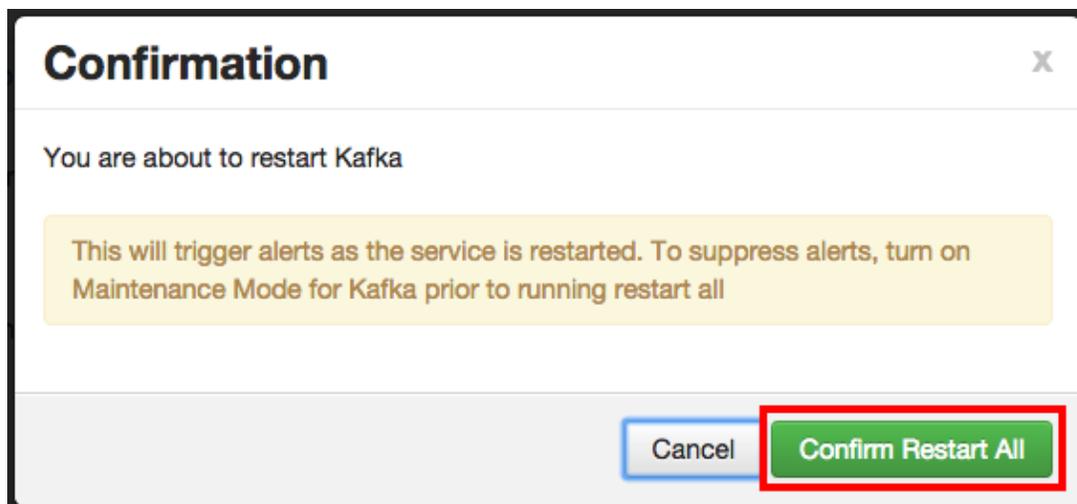


5. Click **OK** on the Save Configuration Changes pop-up.



6. Select **Kafka** in the navigation menu, then select **Restart > Restart All Affected** to restart the Kafka service and load the new configuration.

7. Click **Confirm Restart All** on the confirmation pop-up to confirm the Kafka restart.

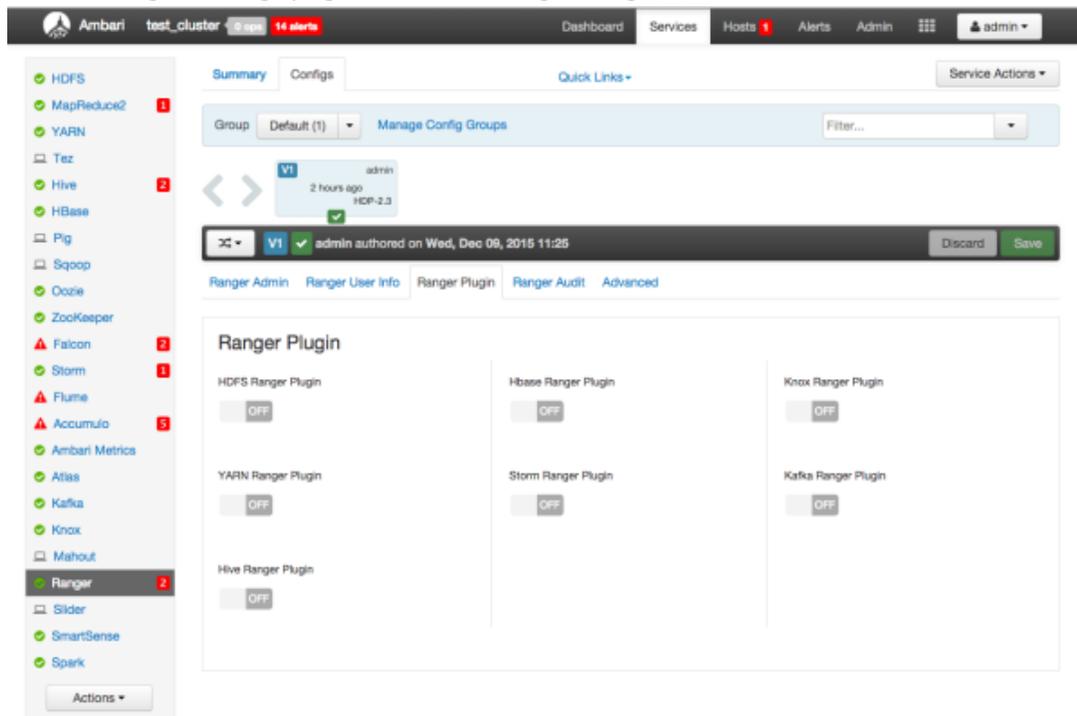


8. After Kafka restarts, the Ranger plugin for Kafka will be enabled.

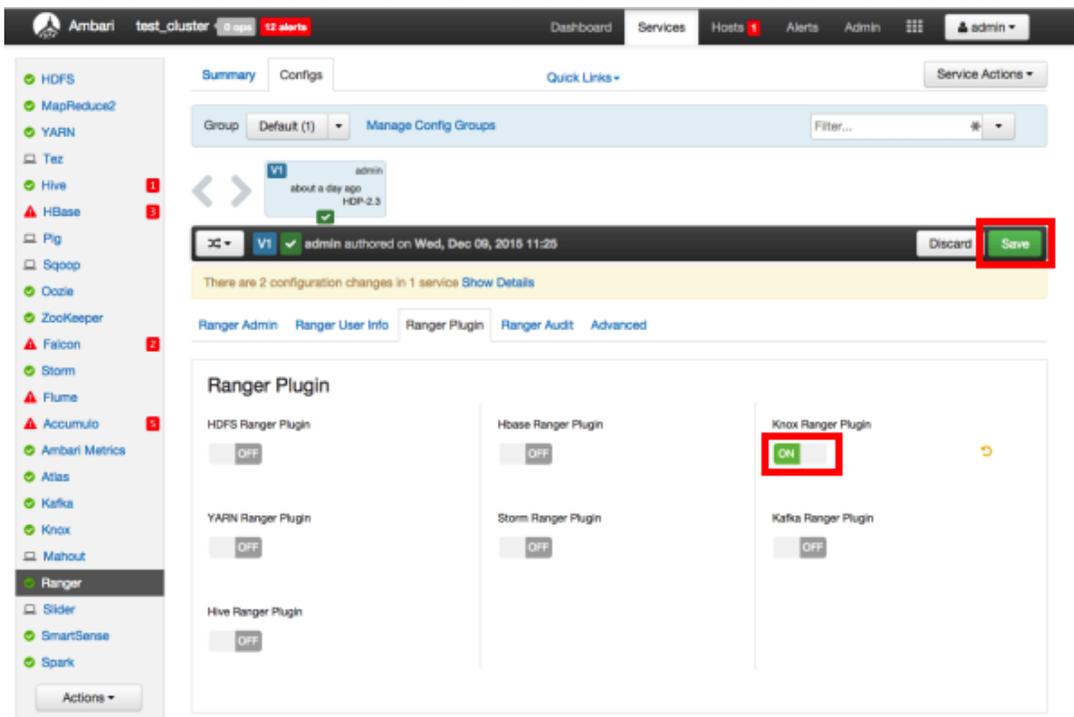
5.5. Knox

Use the following steps to enable the Ranger Knox plugin.

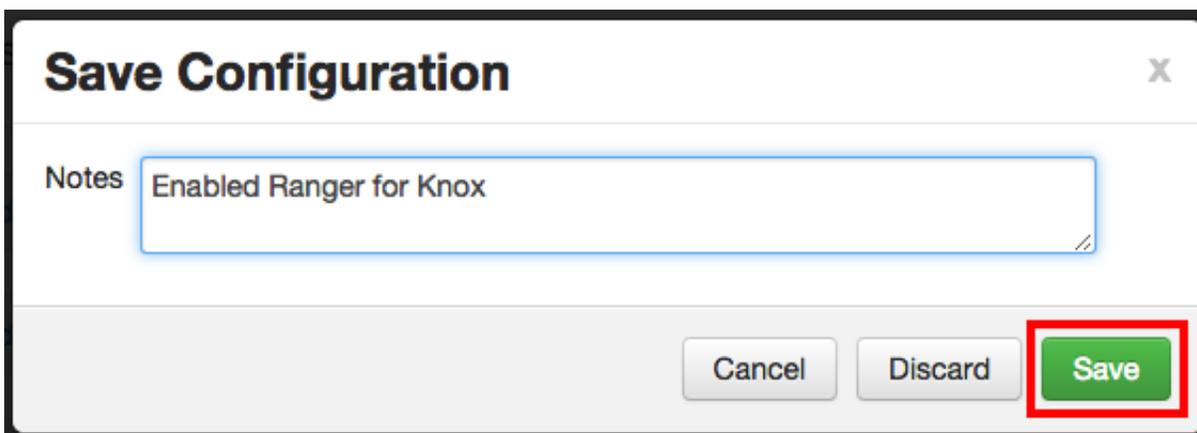
1. On the Ranger Configs page, select the **Ranger Plugin** tab.



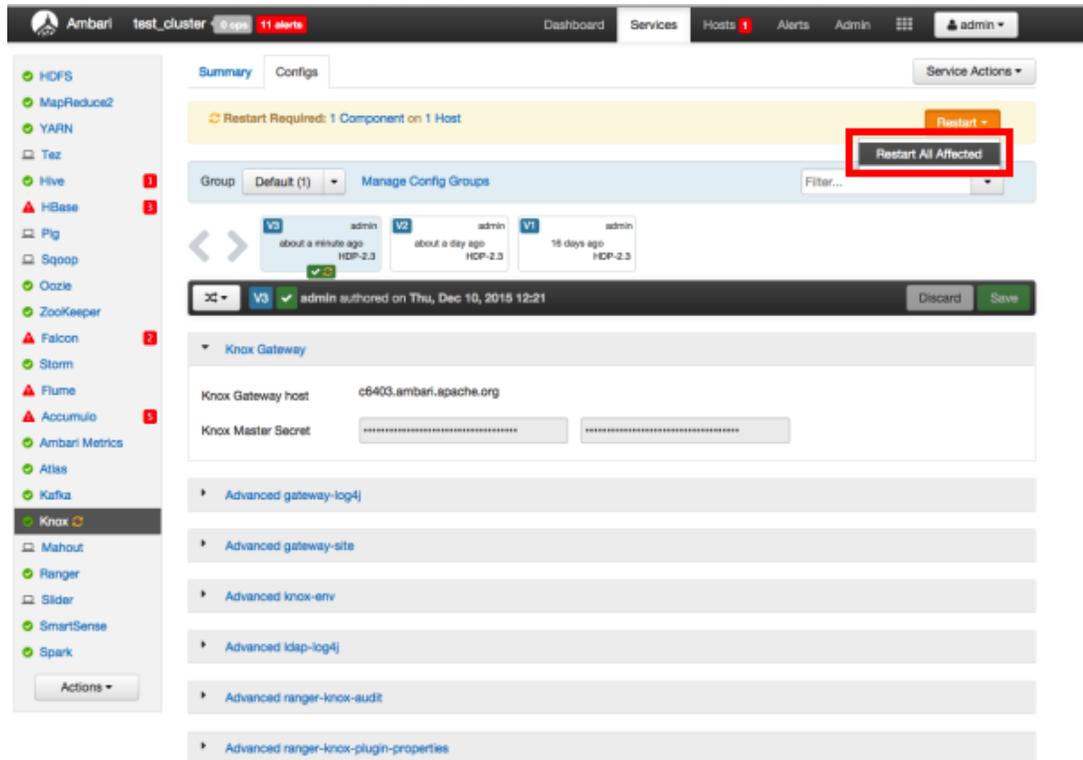
2. Under Knox Ranger Plugin, select **On**, then click **Save** in the black menu bar.



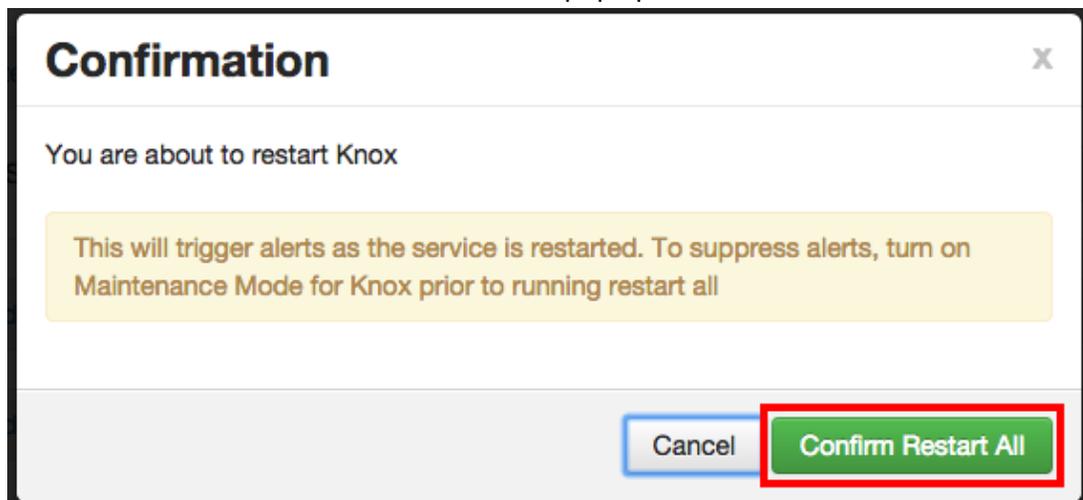
3. A Save Configuration pop-up appears. Type in a note describing the changes you just made, then click **Save**.



4. A Dependent Configuration pop-up appears. Click **OK** to confirm the configuration updates.



7. Click **Confirm Restart All** on the confirmation pop-up to confirm the Knox restart.

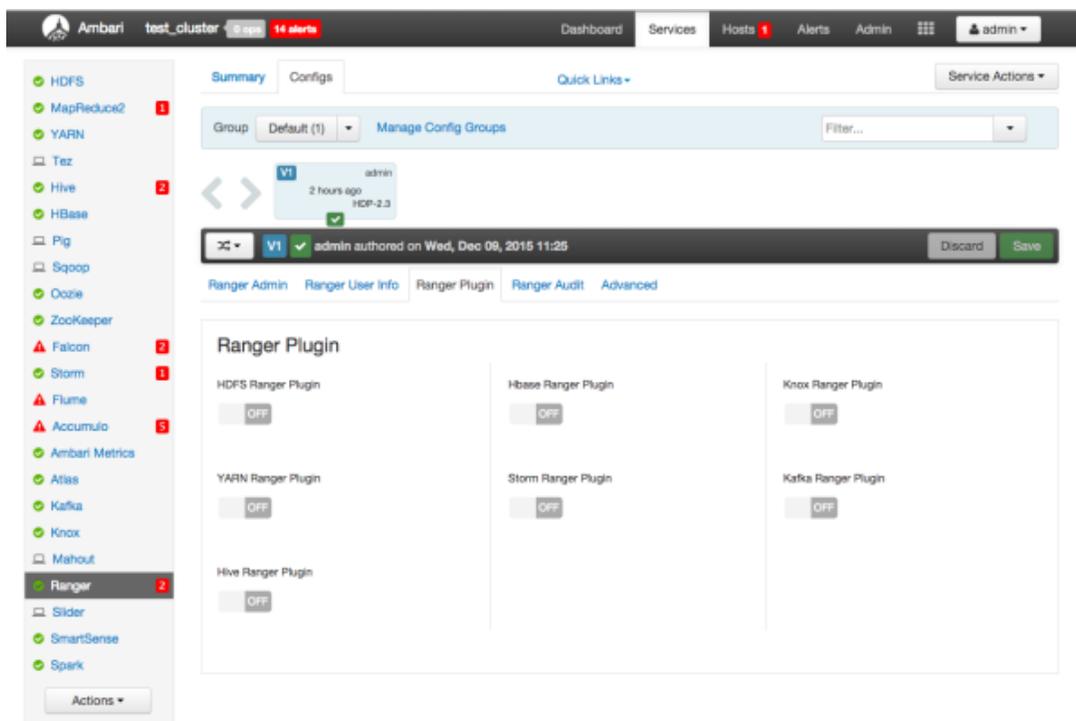


8. After Knox restarts, the Ranger plugin for Knox will be enabled.

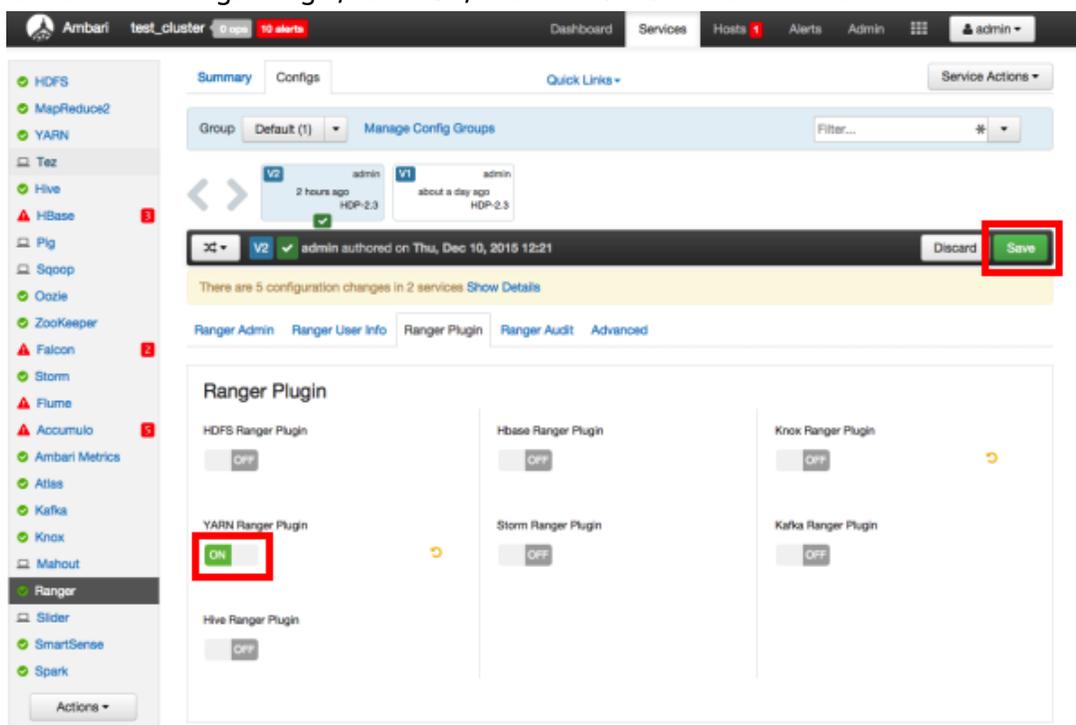
5.6. YARN

Use the following steps to enable the Ranger YARN plugin.

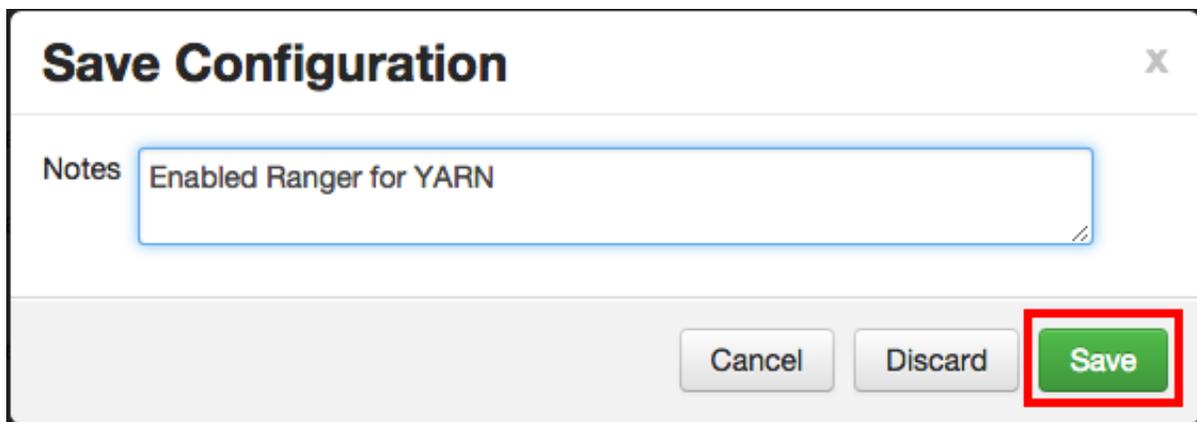
1. On the Ranger Configs page, select the **Ranger Plugin** tab.



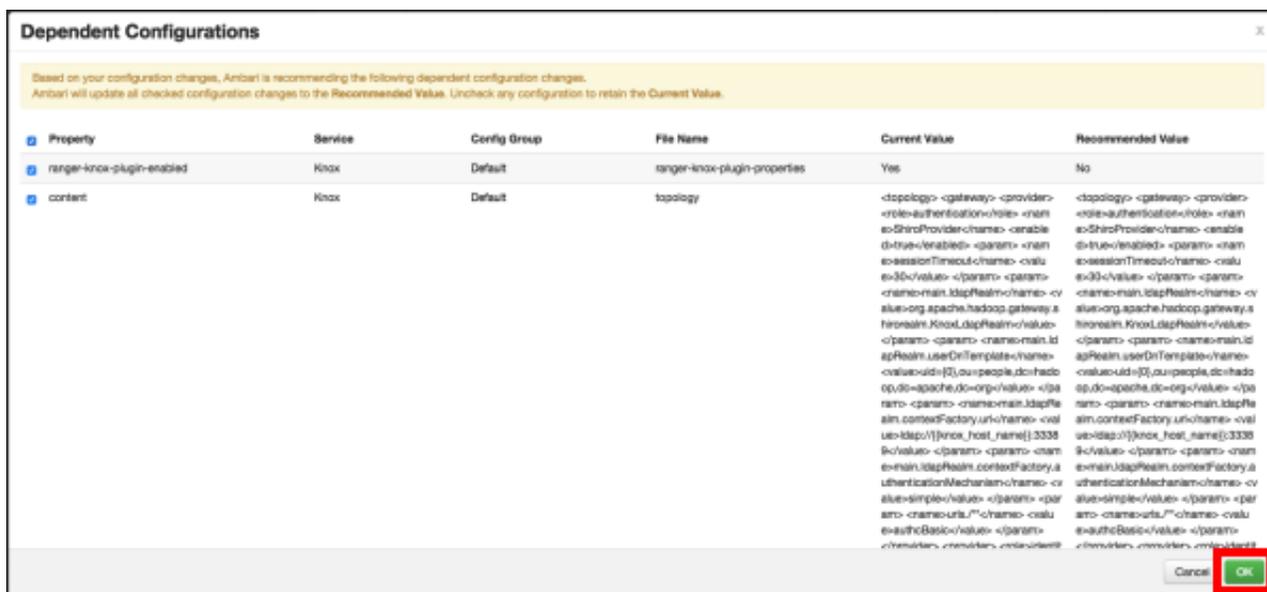
2. Under YARN Ranger Plugin, select On, then click Save in the black menu bar.



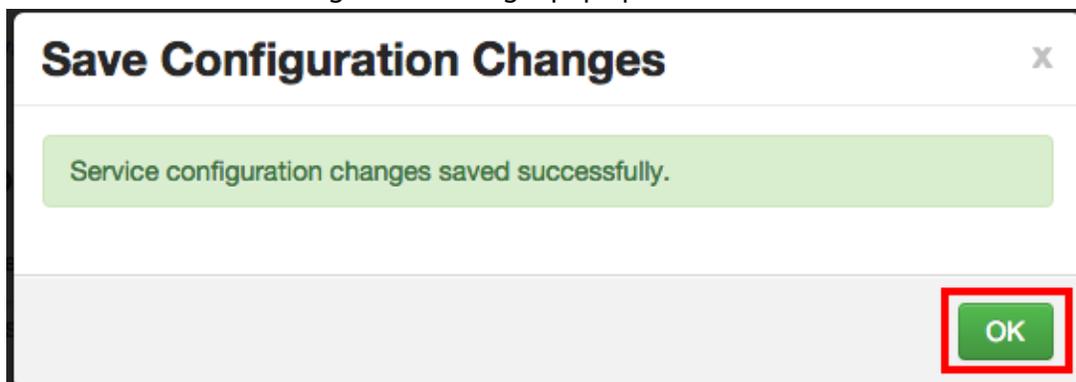
3. A Save Configuration pop-up appears. Type in a note describing the changes you just made, then click Save.



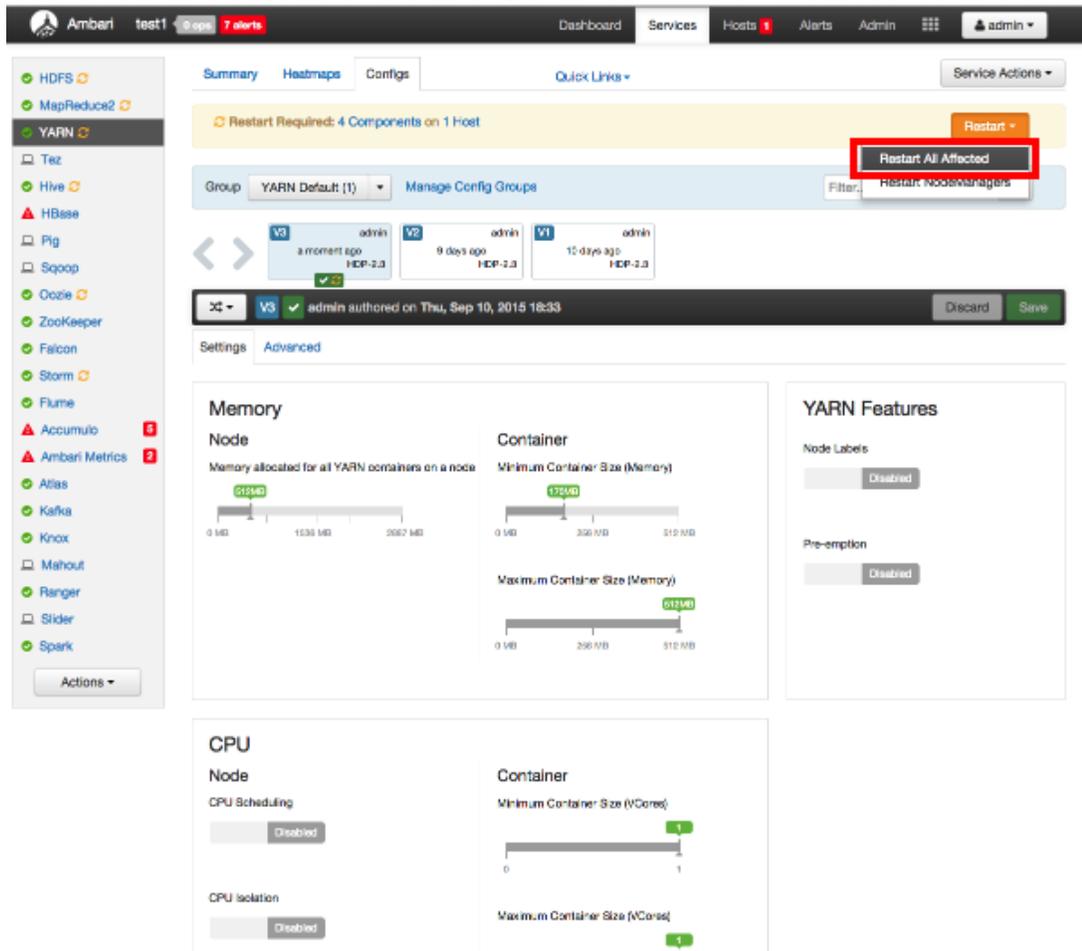
4. A Dependent Configuration pop-up appears. Click **OK** to confirm the configuration updates.



5. Click **OK** on the Save Configuration Changes pop-up.

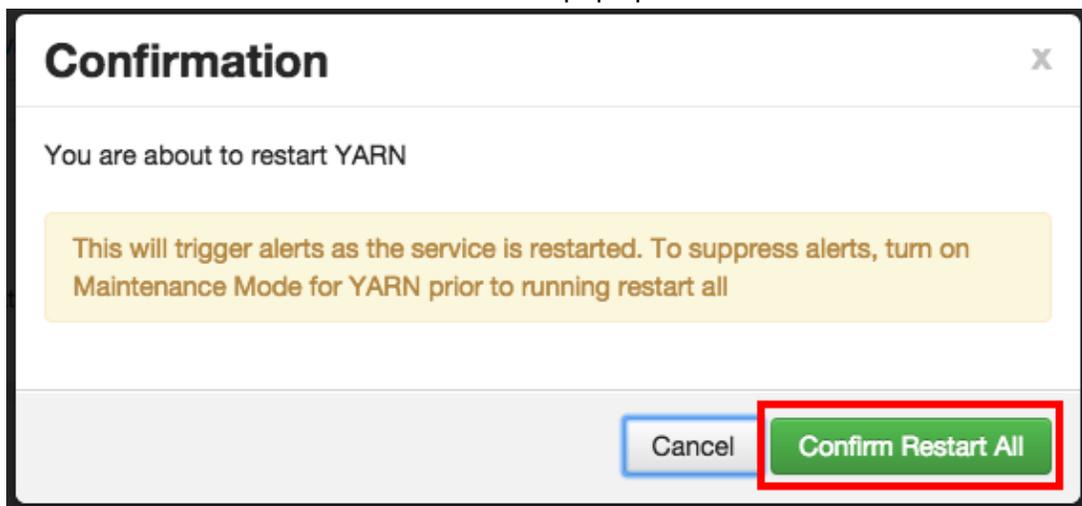


6. Select **YARN** in the navigation menu, then select **Restart > Restart All Affected** to restart the YARN service and load the new configuration.



The screenshot shows the Ambari interface for the YARN service configuration. The left sidebar lists various services, with YARN selected. The main content area shows the YARN configuration page with a yellow banner indicating a restart is required for 4 components on 1 host. A red box highlights the 'Restart All Affected' button in the 'Service Actions' dropdown menu. Below the banner, there are sections for Memory, CPU, and YARN Features, each with sliders and checkboxes for configuration.

7. Click **Confirm Restart All** on the confirmation pop-up to confirm the YARN restart.



The screenshot shows a confirmation dialog box titled 'Confirmation'. The text inside reads: 'You are about to restart YARN'. Below this, a yellow box contains the warning: 'This will trigger alerts as the service is restarted. To suppress alerts, turn on Maintenance Mode for YARN prior to running restart all'. At the bottom right, there are two buttons: 'Cancel' and 'Confirm Restart All', with the latter highlighted by a red box.

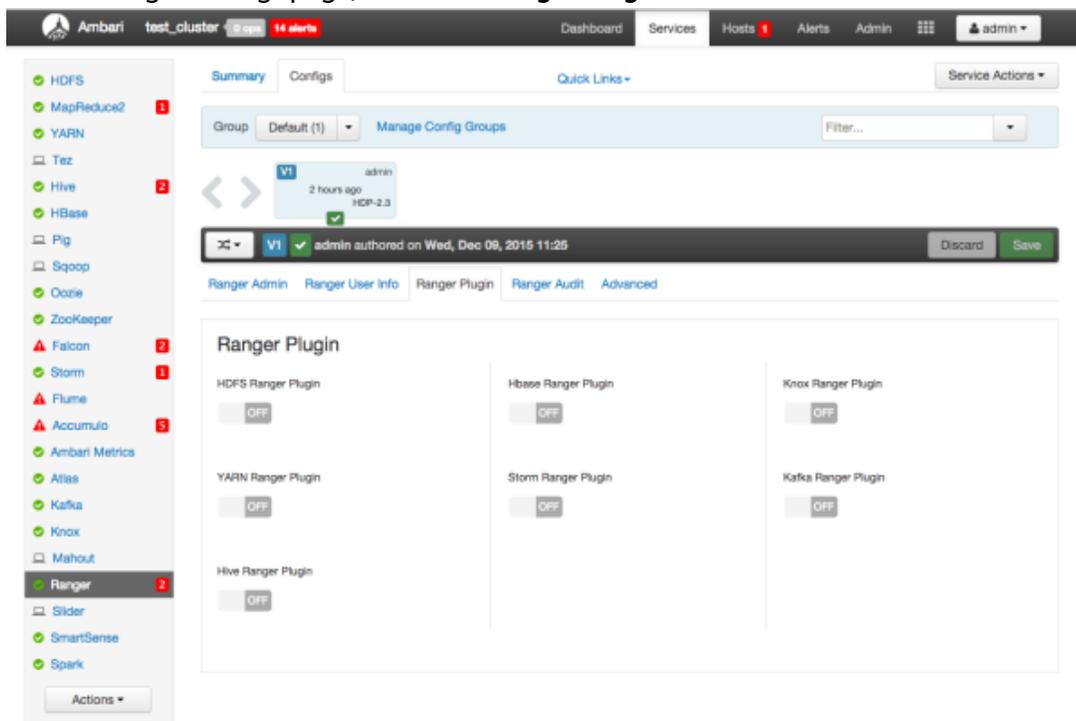
8. After YARN restarts, the Ranger plugin for YARN will be enabled. Other components may also require a restart.

5.7. Storm

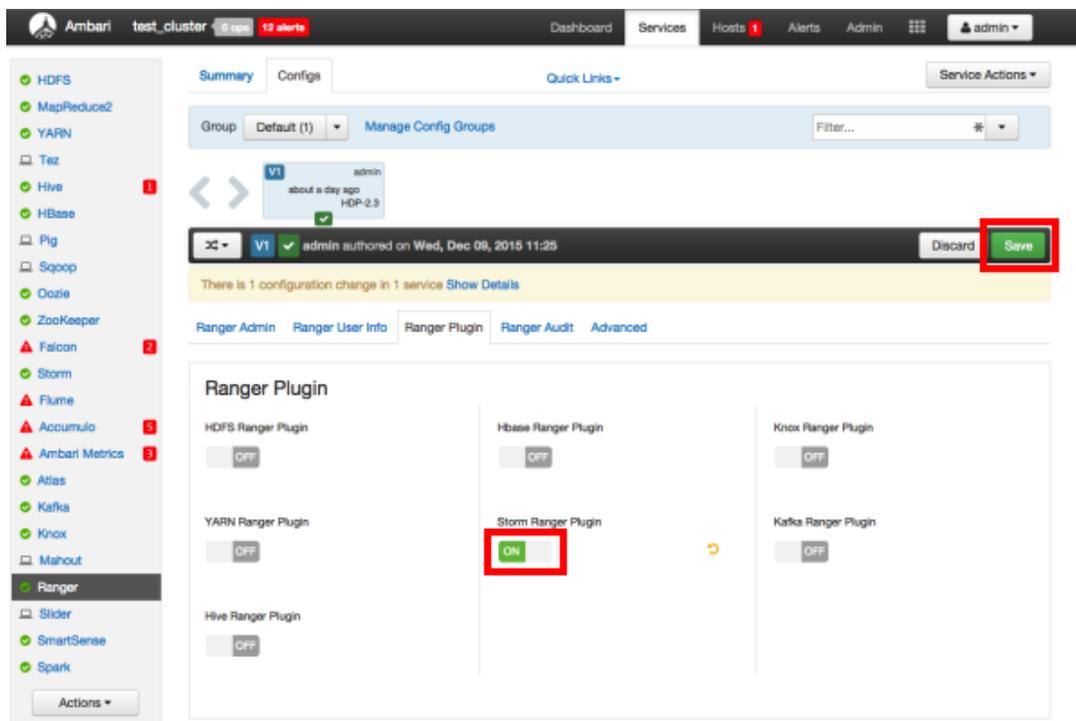
Before you can use the Storm plugin, you must first enable Kerberos on your cluster. To enable Kerberos on your cluster, see [Enabling Kerberos Security](#) in the [Ambari Security Guide](#).

Use the following steps to enable the Ranger Storm plugin.

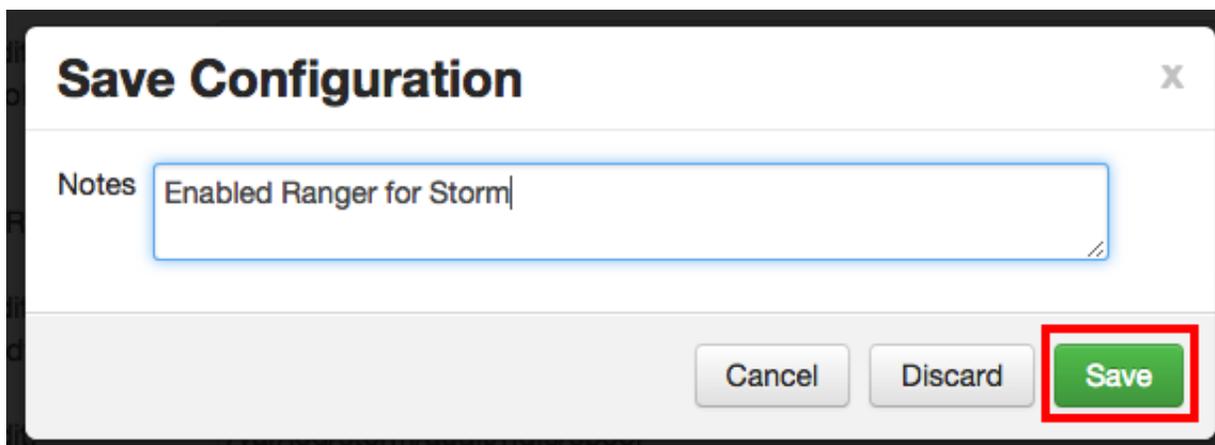
1. On the Ranger Configs page, select the **Ranger Plugin** tab.

The screenshot shows the Ambari web interface for the Ranger configuration page. The top navigation bar includes 'Ambari test_cluster', 'Dashboard', 'Services', 'Hosts', 'Alerts', 'Admin', and a user profile for 'admin'. The left sidebar lists various services, with 'Ranger' highlighted in green and marked with a red '2'. The main content area is titled 'Ranger Configs' and has tabs for 'Summary', 'Configs', and 'Quick Links'. Below the tabs, there's a 'Group' dropdown set to 'Default (1)' and a 'Manage Config Groups' button. A notification bar shows a message from 'admin' 2 hours ago regarding 'HDP-2.3'. Below this is a black menu bar with 'V1' and a checkmark, and a 'Save' button. The 'Ranger Plugin' tab is active, displaying a grid of toggle switches for various plugins: HDFS, YARN, Hive, Hbase, Storm, and Kafka. The 'Storm Ranger Plugin' toggle is currently set to 'OFF'. The 'Ranger Admin' tab is also visible in the sub-navigation.

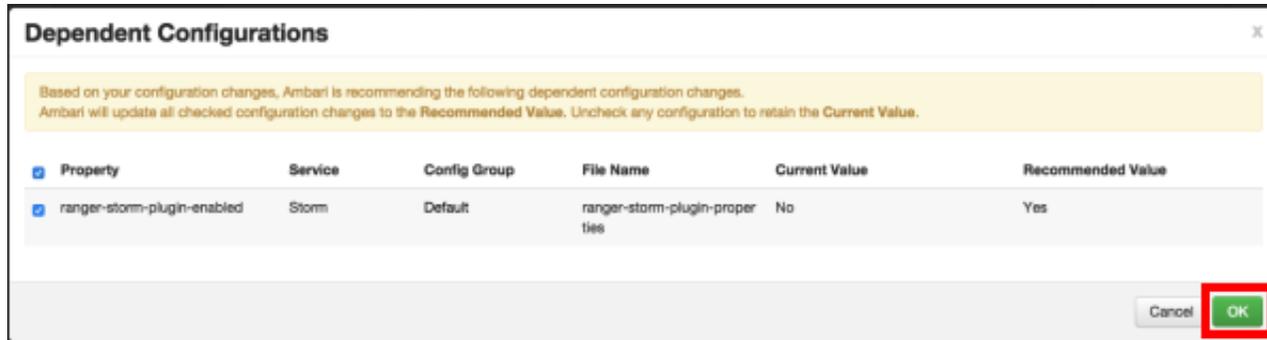
2. Under Storm Ranger Plugin, select **On**, then click **Save** in the black menu bar.



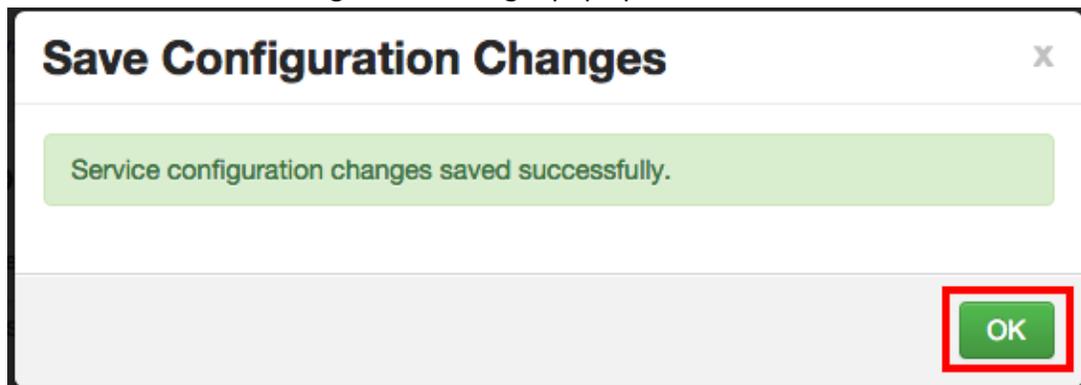
3. A Save Configuration pop-up appears. Type in a note describing the changes you just made, then click **Save**.



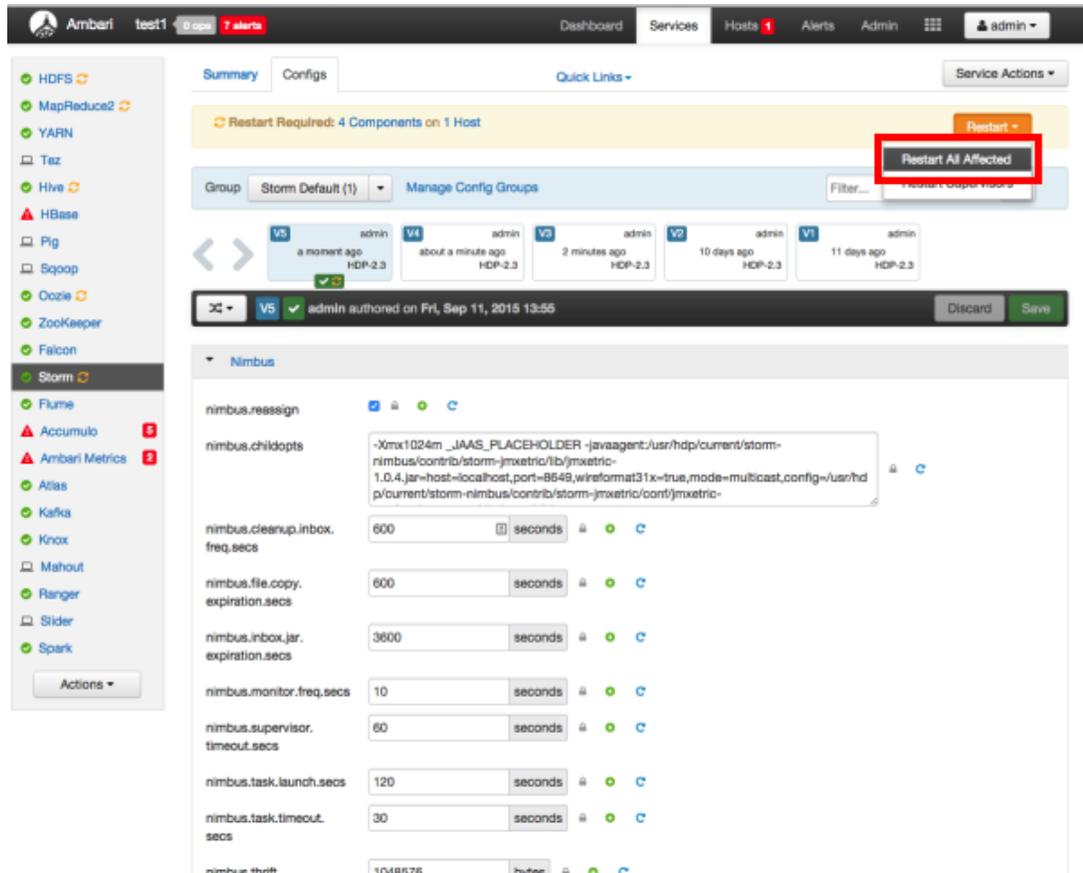
4. A Dependent Configuration pop-up appears. Click **OK** to confirm the configuration updates.



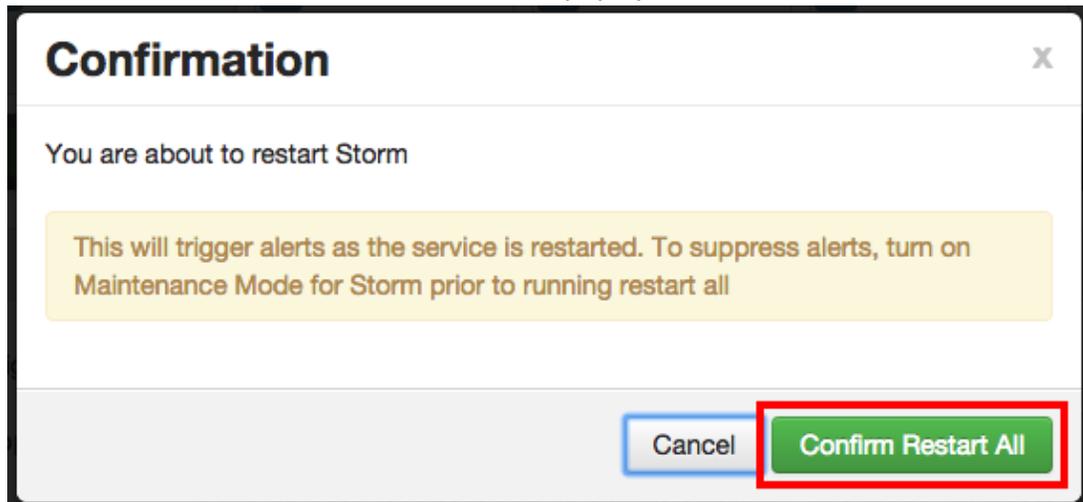
5. Click **OK** on the Save Configuration Changes pop-up.



6. Select **Storm** in the navigation menu, then select **Restart > Restart All Affected** to restart the Storm service and load the new configuration.



7. Click **Confirm Restart All** on the confirmation pop-up to confirm the Storm restart.



8. After Storm restarts, the Ranger plugin for Storm will be enabled.

5.8. Manually Updating HDFS Audit Settings



Note

HDFS audits are enabled by default in the standard Ranger Ambari installation procedure, and are activated automatically when Ranger is enabled for a plugin.

The following steps show how to save Ranger audits to HDFS for HBase. You can use the same procedure for other components.

1. From the Ambari dashboard, select the HBase service. On the Configs tab, scroll down and select **Advanced ranger-hbase-audit**. Select the **Audit to HDFS** check box.
2. Set the HDFS path where you want to store audits in HDFS:

```
xasecure.audit.destination.hdfs.dir = hdfs://  
$NAMENODE_FQDN:8020/ranger/audit
```

Refer to the `fs.defaultFS` property in the **Advanced core-site** settings.



Note

For NameNode HA, `NAMENODE_FQDN` is the cluster name. In order for this to work, `/etc/hadoop/conf/hdfs-site.xml` needs to be linked under `/etc/<component_name>/conf`.

3. Enable the Ranger plugin for HBase.
4. Make sure that the plugin sudo user should has permission on the HDFS Path:

```
hdfs://NAMENODE_FQDN:8020/ranger/audit
```

For example, we need to create a Policy for Resource : `/ranger/audit`, all permissions to user `hbase`.

5. Save the configuration updates and restart HBase.
6. Generate some audit logs for the HBase component.
7. Check the HDFS component logs on the NameNode:

```
hdfs://NAMENODE_FQDN:8020/ranger/audit
```



Note

For a secure cluster, use the following steps to test audit to HDFS for STORM/KAFKA/KNOX:

- In `core-site.xml` set the `hadoop.proxyuser.<component>.groups` property with value `" * "` or service user.

- For the Knox plugin there is one additional property to add to core-site.xml. Add `hadoop.proxyuser.<component>.users` property with value “ * ” or service user (i.e Knox).
- Link to `/etc/hadoop/conf/core-site.xml` under `/etc/<component_name>/conf`.
- Verify the service user principal.
- Make sure that the component user has permissions on HDFS.

5.9. Manually Updating Solr Audit Settings

You can save and store Ranger audits to Solr if you have installed and configured the Solr service in your cluster.

It is recommended that Ranger audits be written to both Solr and HDFS. Audits to Solr are primarily used to enable queries from the Ranger Admin UI. HDFS is a long-term destination for audits – audits stored in HDFS can be exported to any SIEM system, or to another audit store.



Note

If you enabled Solr Audits as part of the standard Ambari installation procedure, audits to Solr are activated automatically when Ranger is enabled for a plugin.

To save Ranger audits to Solr:

1. From the Ambari dashboard, select the Ranger service. On the Configs tab, scroll down and select **Advanced ranger-admin-site**. Set the following property values:
 - `ranger.audit.source.type = solr`
 - `ranger.audit.solr.urls = http://solr_host:6083/solr/ranger_audits`
 - `ranger.audit.solr.username = ranger_solr`
 - `ranger.audit.solr.password = NONE`
2. Restart the Ranger service.
3. After the Ranger service has been restarted, you will then need to make specific configuration changes for each plugin to ensure that the plugin's data is captured in Solr.
4. For example, if you would like to configure HBase for audits to Solr, perform the following steps:
 - Select the Audit to Solr checkbox in Advanced ranger-hbase-audit.
 - Enable the Ranger plugin for HBase.

- Restart the HBase component.
5. Verify that the Ranger audit logs are being passed to Solr by opening one of the following URLs in a web browser:

`http://{RANGER_HOST_NAME}:6080/index.html#!/reports/audit/bigData`

`http://{SOLR_HOST}:6083/solr/ranger_audits`

6. Ranger Plugins - Kerberos Overview

If you are using a Kerberos-enabled cluster, there are a number of steps you need to follow to ensure you can use the different Ranger plugins on a Kerberos cluster. These plugins are:

1. [HDFS \[74\]](#)
2. [Hive \[75\]](#)
3. [HBase \[75\]](#)
4. [Knox \[76\]](#)

6.1. HDFS

To enable the Ranger HDFS plugin on a Kerberos-enabled cluster, perform the steps described below.

1. Create the system (OS) user `rangerhdfslookup`. Make sure this user is synced to Ranger Admin (under *users/groups* tab in the Ranger Admin User Interface).
2. Create a Kerberos principal for `rangerhdfslookup` by entering the following command:

```
• kadmin.local -q 'addprinc -pw rangerhdfslookup
  rangerhdfslookup@example.com'
```



Note

A single user/principal (e.g., `rangerrepouser`) can also be created and used across services.

3. Navigate to the HDFS service.
4. Click on the **Config** tab.
5. Navigate to *advanced ranger-hdfs-plugin-properties* and update the properties listed in the table shown below.

Property Name	Value
Enable Ranger for HDFS	<input checked="" type="checkbox"/>
Audit to HDFS	<input checked="" type="checkbox"/>
Audit to DB	<input checked="" type="checkbox"/>
policy User for HDFS	ambari-qa
Ranger repository config password	hdfsuser
Ranger repository config user	rangerhdfslookup@EXAMPLE.COM
common name for certificate	
hadoop.rpc.protection	*
SSL_KEYSTORE_FILE_PATH	/etc/hadoop/conf/ranger-plugin-keystore.jks

Table 6.1. HDFS Plugin Properties

Configuration Property Name	Value
Ranger repository config user	rangerhdfslookup@example.com

Configuration Property Name	Value
Ranger repository config password	rangerhdfslookup
common.name.for.certificate	blank

6. After updating these properties, click **Save** and restart the HDFS service.

6.2. Hive



Important

You should not use the Hive CLI after enabling the Ranger Hive plugin. The Hive CLI is not supported in HDP-2.2.0 and higher versions, and may break the install or lead to other unpredictable behavior. Instead, you should use the [HiveServer2 Beeline CLI](#).

To enable the Ranger HBase plugin on a Kerberos-enabled cluster, perform the steps described below.

1. Create the system (OS) user `rangerhivelookup`. Make sure this user is synced to Ranger Admin (under *users/groups* tab in the Ranger Admin UI).
2. Create a Kerberos principal for `rangerhivelookup` by entering the following command:
 - `kadmin.local -q 'addprinc -pw rangerhivelookup rangerhivelookup@example.com'`
3. Navigate to the Hive service.
4. Click on the **Config** tab and navigate to *advanced ranger-hive-plugin-properties*.
5. Update the following properties with the values listed in the table below.

Table 6.2. Hive Plugin Properties

Configuration Property Name	Value
Ranger repository config user	rangerhivelookup@example.com
Ranger repository config password	rangerhivelookup
common.name.for.certificate	blank

6. After updating these properties, click **Save** and then restart the Hive service.

6.3. HBase

To enable the Ranger HBase plugin on a Kerberos-enabled cluster, perform the steps described below.

1. Create the system (OS) user `rangerhbaselookup`. Make sure this user is synced to Ranger Admin (under *users/groups* tab in the Ranger Admin UI).
2. Create a Kerberos principal for `rangerhbaselookup` by entering the following command:

- `kadmin.local -q 'addprinc -pw rangerhbaselookup rangerhbaselookup@example.com'`

3. Navigate to the HBase service.
4. Click on the **Config** tab and go to *advanced ranger-hbase-plugin-properties*.
5. Update the following properties with the values listed in the table below.

Table 6.3. HBase Plugin Properties

Configuration Property Name	Value
Ranger repository config user	rangerhbaselookup@example.com
Ranger repository config password	rangerhbaselookup
common.name.for.certificate	blank

6. After updating these properties, click **Save** and then restart the HBase service.

6.4. Knox

To enable the Ranger Knox plugin on a Kerberos-enabled cluster, perform the steps described below.

1. Create the system (OS) user `rangerknoxlookup`. Make sure this user is synced to Ranger Admin (under *users/groups* tab in the Ranger Admin UI).
2. Create a Kerberos principal for `rangerknoxlookup` by entering the following command:
 - `kadmin.local -q 'addprinc -pw rangerknoxlookup rangerknoxlookup@example.com'`
3. Navigate to the Knox service.
4. Click on the **Config** tab and navigate to *advanced ranger-knox-plugin-properties*.
5. Update the following properties with the values listed in the table below.

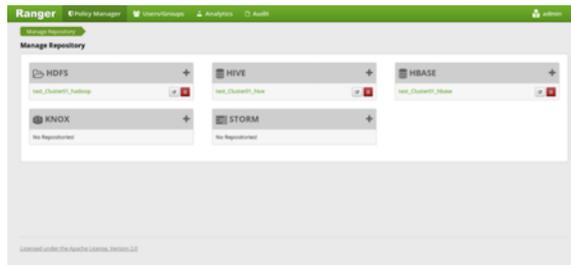
Table 6.4. Knox Plugin Properties

Configuration Property Name	Value
Ranger repository config user	rangerknoxlookup@example.com
Ranger repository config password	rangerknoxlookup
common.name.for.certificate	blank

6. After updating these properties, click **Save** and then restart the Knox service.
7. Open the Ranger Admin UI by entering the following information:
 - `http://ranger-host>:6080`

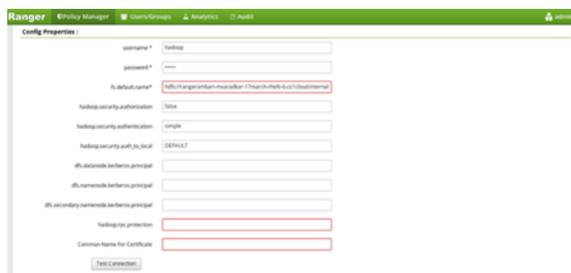
- **username/password** - *admin/admin*. or use *username* as shown in *advanced ranger-env* under the **Config** tab of the Ranger service, and *password* as shown in **Admin Settings**.
8. After you have successfully logged into the system, you will be redirected to the Policy Manager page.

Figure 6.1. Knox Policy Manager



9. Click on the repository (clusterName_hadoop) **Edit** option under the HDFS box.

Figure 6.2. Knox Repository Edit



10. Update the following properties listed in the table below under the Config Properties section:

Table 6.5. Knox Configuration Properties

Configuration Property Name	Value
fs.default.name	hdfs
hadoop.rpc.protection	blank
common.name.for.certificate	blank

11. Click on **Named Test Connection**. You should see a *Connected Successfully* dialog box appear.
12. Click **Save**.